

Salvatore Nicosia
Bruno Siciliano
Antonio Bicchi
Paolo Valigi (Eds.)

RAMSETE

Articulated and Mobile Robotics
for Services and Technologies



Springer

Lecture Notes in Control and Information Sciences 270

Editors: M. Thoma · M. Morari

Springer

Berlin

Heidelberg

New York

Barcelona

Hong Kong

London

Milan

Paris

Singapore

Tokyo

Engineering  **ONLINE LIBRARY**
<http://www.springer.de/engine/>

Salvatore Nicosia, Bruno Siciliano,
Antonio Bicchi, Paolo Valigi (Eds)

RAMSETE

**Articulated and Mobile Robotics
for Services and Technologies**



Springer

Series Advisory Board

A. Bensoussan · P. Fleming · M.J. Grimble · P. Kokotovic ·
A.B. Kurzhanski · H. Kwakernaak · J.L. Massey

Editors

Professor Salvatore Nicosia
Dipartimento di Informatica Sistemi
e Produzione
Università di Roma "Tor Vergata"
Via Tor Vergata 110
00133 Roma, Italy

Professor Bruno Siciliano
Dipartimento di Informatica
e Sistemistica
Università di Napoli Federico II
Via Claudio 21
80125 Napoli, Italy

Professor Antonio Bicchi
Centro "E. Piaggio"
Università di Pisa
Via Diotisalvi 2
56126 Pisa, Italy

Professor Paolo Valigi
Dipartimento di Ingegneria Elettronica
e dell'Informazione
Università di Perugia
Via G. Duranti 93
06125 Perugia, Italy

Cataloging-in-Publication Data applied for
Die Deutsche Bibliothek – CIP-Einheitsaufnahme
RAMSETE: articulated and mobile robotics for services and technologies / Salvatore Nicosia...(ed)
Berlin; Heidelberg; New York; Barcelona; Hong Kong; London; Milano; Paris; Singapore; Tokyo:
Springer, 2001
(Lecture Notes in control and information sciences; 270)
(Engineering online library)
ISBN 3-540-42090-8

ISBN 3-540-42090-8 Springer-Verlag Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilm or in other ways, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer-Verlag. Violations are liable for prosecution act under German Copyright Law.

Springer-Verlag Berlin Heidelberg New York
a member of BertelsmannSpringer Science + Business Media GmbH

<http://www.springer.de>

© Springer-Verlag Berlin Heidelberg 2001
Printed in Germany

The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Typesetting: Digital data supplied by author. Data-conversion by PTP-Berlin, Stefan Sossna
Cover-Design: design & production GmbH, Heidelberg
Printed on acid-free paper SPIN 10789169 62/3020Rw - 5 4 3 2 1 0

Preface

Robotics applications, initially developed for industrial and manufacturing contexts, are now strongly present in several fields. Besides well-known space and high-technology applications, robotics for every day life and medical services is becoming more and more popular. As an example, robotic manipulators are particularly useful in surgery and radiation treatments, they could be employed for civil demining, for helping disabled people, and ultimately for domestic tasks, entertainment and education. Such a kind of robotic applications require the integration of many different skills. Autonomous vehicles and mobile robots in general must be integrated with articulated manipulators. Many robotic technologies (sensors, actuators and computing systems) must be properly used with specific technologies (localisation, planning and control technologies). The task of designing robots for these applications is a hard challenge: a specific competence in each area is demanded, in the effort of a truly integrated multidisciplinary design.

This monograph stems out of the research project RAMSETE (Robotica Articolata e Mobile per i SErvizi e le TEcnologie — Articulated and Mobile Robots for SERVICES and TEchnologies), funded in 1999–2000 by the Italian Ministry for University, Scientific and Technology Research (MURST), involving a significant portion of the Italian robotics community; namely, the research groups at: University of Bologna, University of Genoa, Polytechnic of Milan, University of Naples, University of Perugia, University of Pisa, University of Rome “La Sapienza”, University of Rome “Tor Vergata”, Third University of Rome, and Polytechnic of Turin. At the web site <http://www-lar.deis.unibo.it/ramsete> a complete description of the project is available.

Early research activities in Italy in the field of robotics date back to the late fifties. At that time, simultaneously with other activities in the United States, an extremely versatile master/slave tele-manipulator was under development at CNEN laboratories: the MASCOT. Some of the new versions of such manipulator are still in operation at the Joint European Torus laboratory, United Kingdom, where fusion reactors are under study. Since that time, the Italian scientific community has always played a major role in the robotics field, with a new considerable interest about the end of the seventies when, downstream of the applications in manufacturing, some components of the Italian academy started giving contributions of relevant scientific interest in the field of articulated robotics for industrial applications in structured environments. Their contributions to the development of modern robotics were recognised by the whole international community. Among many others, relevant results have been given in the areas of dynamic modelling and control of articulated robots, adaptive and learning control, modelling and control

of elastic robots, asymptotic state observers, design and realisation of end effectors.

The main objective of this monograph is to provide robotic researchers and developers with: a) a reliable and authoritative source of information, b) meaningful and feasible developments, c) significance and impact of new problems in the field. The monograph emphasises all the implementation aspects related to major robot subsystems, i.e. both technology structures and specific methodologies. The study of robotic subsystems aimed at the integration in a single complex system, with autonomous motion capabilities, represents the current frontier of scientific research in the field of robotics. In particular, the most relevant open problems deal with the integration of functions and autonomous decision capabilities of articulated robots, such as control of co-operating arms, their interaction with the environment (contact and impact phenomena), as well as the integration of functions and autonomous decision capabilities of mobile robots, such as those related to localisation, planning of tasks and motion, and motion control. In addition, in all these situations the following factors attain a crucial importance: hardware/software architectures for computation and control, availability of advanced actuators and sensory systems.

The technical issues arising in the study of the above problems are examined in the perspective of the required skills. In particular, there are some technological and methodological skills, that are necessary for the effective realisation of robots for field and service applications. In this monograph, these skills have been arranged into six methodological areas, three of them focused on articulated robots and the others on mobile/autonomous robots, and four technological areas. The areas are listed below and the research group leading the activities during the project within each area is indicated.

– Technological areas

Hardware/software architectures (Univ. Genoa)
 Joint actuators (Polytech. Milan)
 End effectors (Univ. Pisa)
 Sensors (Univ. Bologna)

– Methodological areas

Articulated robots

Flexible link control (Univ. Perugia)
 Interaction control (Univ. Naples)
 Impact modelling and control (Polytech. Turin)

Mobile and autonomous robots

Vehicle control (Univ. Rome “La Sapienza”)
 Planning (Third Univ. Rome)
 Localisation (Univ. Rome “Tor Vergata”)

The book is intended for graduate students, researchers, scientists and scholars who wish to broaden and strengthen their knowledge in robotics.

Besides thanking all the Authors for their valuable contributions to this monograph, we wish to extend our appreciation to all the participants to the RAMSETE project who have produced significant research results during the latest two years. Warmest thanks are for Nicholas Pinfield, Engineering Editor, and his assistants Hannah Ransley and Oliver Jackson of Springer-Verlag, London. A final word of thanks goes to Costanzo Manes for the pictorial illustration below.

February 2001

*Salvatore Nicosia
Bruno Siciliano
Antonio Bicchi
Paolo Valigi*



MURST

List of Contributors

Damiano Angeletti

Dipartimento di Informatica Sistemistica e Telematica
Università degli Studi di Genova
Via Opera Pia 13
16145 Genova, Italy
henoch@dist.unige.it

Antonio Bicchi

Centro “E. Piaggio”
Università degli Studi di Pisa
Via Diotisalvi 2
56126 Pisa, Italy
bicchi@ing.unipi.it

Tommaso Bozzo

Dipartimento di Informatica Sistemistica e Telematica
Università degli Studi di Genova
Via Opera Pia 13
16145 Genova, Italy
bozzo@dist.unige.it

Fabrizio Caccavale

PRISMA Lab, Dipartimento di Informatica e Sistemistica
Università degli Studi di Napoli Federico II
Via Claudio 21
80125 Napoli, Italy
caccaval@unina.it

Giorgio Cannata

Dipartimento di Informatica Sistemistica e Telematica
Università degli Studi di Genova
Via Opera Pia 13
16145 Genova, Italy
cannata@dist.unige.it

Giuseppe Casalino

Dipartimento di Informatica Sistemistica e Telematica
Università degli Studi di Genova
Via Opera Pia 13
16145 Genova, Italy
pino@dist.unige.it

Alessandro De Luca

Dipartimento di Informatica e Sistemistica
Università degli Studi di Roma “La Sapienza”
Via Eudossiana 18
00184 Roma, Italy
deluca@dis.uniroma1.it

Gianni Ferretti

Dipartimento di Elettronica e Informazione
Politecnico di Milano
Piazza Leonardo da Vinci 32
20133 Milano, Italy
ferretti@elet.polimi.it

Antonio Ficola

Dipartimento di Ingegneria Elettronica e dell’Informazione
Università degli Studi di Perugia
Via G. Duranti 93
06125 Perugia, Italy
ficola@diei.unipg.it

Mario Luca Fravolini

Dipartimento di Ingegneria Elettronica e dell’Informazione
Università degli Studi di Perugia
Via G. Duranti 93
06125 Perugia, Italy
fravolini@diei.unipg.it

Fabrizio Grandoni

Dipartimento di Informatica Sistemi e Produzione
Università degli Studi di Roma “Tor Vergata”
Via Tor Vergata 110
00133 Roma, Italy
grandoni@disp.uniroma2.it

Marina Indri

Dipartimento di Automatica e Informatica
Politecnico di Torino
Corso Duca degli Abruzzi 24
10129 Torino, Italy
indri@polito.it

Vijay Kumar

Grasp Laboratory
University of Pennsylvania
Room 111, Towne Building
220 S. 33rd Street
Philadelphia, PA 19104, USA
`kumar@central.cis.upenn.edu`

Michele La Cava

Dipartimento di Ingegneria Elettronica e dell'Informazione
Università degli Studi di Perugia
Via G. Duranti 93
06125 Perugia, Italy
`lacava@diei.unipg.it`

GianAntonio Magnani

Dipartimento di Elettronica e Informazione
Politecnico di Milano
Piazza Leonardo da Vinci 32
20133 Milano, Italy
`magnani@elet.polimi.it`

Agostino Martinelli

Dipartimento di Informatica Sistemi e Produzione
Università degli Studi di Roma "Tor Vergata"
Via Tor Vergata 110
00133 Roma, Italy
`ago@robot2.disp.uniroma2.it`

Francesco Martinelli

Dipartimento di Informatica Sistemi e Produzione
Università degli Studi di Roma "Tor Vergata"
Via Tor Vergata 110
00133 Roma, Italy
`martinelli@disp.uniroma2.it`

Claudio Melchiorri

LAR, Dipartimento di Elettronica Informatica Sistemistica
Università degli Studi di Bologna
Viale Risorgimento 2
40136 Bologna, Italy
cmelchiorri@deis.unibo.it

Alessandro Micarelli

Dipartimento di Informatica e Automazione
Università degli Studi di Roma Tre
Via della Vasca Navale 79
00146 Roma, Italy
micarelli@dia.uniroma3.it

Ciro Natale

PRISMA Lab, Dipartimento di Informatica e Sistemistica
Università degli Studi di Napoli Federico II
Via Claudio 21
80125 Napoli, Italy
cinatale@unina.it

Salvatore Nicosia

Dipartimento di Informatica Sistemi e Produzione
Università degli Studi di Roma “Tor Vergata”
Via Tor Vergata 110
00133 Roma, Italy
nicosia@disp.uniroma2.it

Giuseppe Oriolo

Dipartimento di Informatica e Sistemistica
Università degli Studi di Roma “La Sapienza”
Via Eudossiana 18
00184 Roma, Italy
oriolo@dis.uniroma1.it

Stefano Panzieri

Dipartimento di Informatica e Automazione
Università degli Studi di Roma Tre
Via della Vasca Navale 79
00146 Roma, Italy
panzieri@dia.uniroma3.it

Paolo Rocco

Dipartimento di Elettronica e Informazione
Politecnico di Milano
Piazza Leonardo da Vinci 32
20133 Milano, Italy
rocco@elet.polimi.it

Lorenzo Sciavicco

Dipartimento di Informatica e Automazione
Università degli Studi di Roma Tre
Via della Vasca Navale 79
00146 Roma, Italy
sciavicco@unina.it

Bruno Siciliano

PRISMA Lab, Dipartimento di Informatica e Sistemistica
Università degli Studi di Napoli Federico II
Via Claudio 21
80125 Napoli, Italy
siciliano@unina.it

Giovanni Ulivi

Dipartimento di Informatica e Automazione
Università degli Studi di Roma Tre
Via della Vasca Navale 79
00146 Roma, Italy
ulivi@dia.uniroma3.it

Paolo Valigi

Dipartimento di Ingegneria Elettronica e dell'Informazione
Università degli Studi di Perugia
Via G. Duranti 93
06125 Perugia, Italy
valigi@diei.unipg.it

Marilena Vendittelli

Dipartimento di Informatica e Sistemistica
Università degli Studi di Roma "La Sapienza"
Via Eudossiana 18
00184 Roma, Italy
venditt@dis.uniroma1.it

Luigi Villani

PRISMA Lab, Dipartimento di Informatica e Sistemistica
Università degli Studi di Napoli Federico II
Via Claudio 21
80125 Napoli, Italy
`lvillani@unina.it`

Table of Contents

List of Contributors	xiv
-----------------------------------	-----

Strategies for Control and Coordination within Multiarm Systems

Giuseppe Casalino, Damiano Angeletti, Tommaso Bozzo, and Giorgio Cannata	1
1 Introduction	1
2 The Functional Architecture	3
3 An Example of Supporting Hw and Sw Architecture	6
4 The Low Level Control Layer	8
5 The Medium Level Control Layer	11
5.1 The Teleoperated/Automatic Independent Motion Control of the Arms	12
5.2 The Teleoperated/Automatic Coordinated Motion Control of the Arms	15
5.3 Interaction Control during Coordinated Motion	21
6 Comments on the Role of the High Level Control Layer	23
7 Conclusions	24

Modelling and Control of Servomechanisms

Gianni Ferretti, GianAntonio Magnani, and Paolo Rocco	27
1 Introduction	27
2 Modelling and Compensation of Torque Disturbances	29
2.1 Torque Disturbance Generation	29
2.2 Existing Solutions for Torque Minimisation	30
2.3 Torque Disturbance Model	31
2.4 Spectral Identification of the Disturbance	31
2.5 Adaptive Identification of the Disturbance	33
2.6 Design of a Compensation Technique	34
2.7 Experimental Results	35
3 Modelling and Compensation of Load Oscillations	37
3.1 Model of an Elastic Servo Mechanism	37
3.2 Existing Solutions for the Compensation of Load Oscillations ..	39
3.3 Classical Control and Load Oscillations	39

4	LQG Load Position Control	45
4.1	Design of the Optimal Controller.....	46
4.2	Design of the Kalman Filter	48
4.3	Design of the Feedforward Filters	48
4.4	Experimental Results.....	50
5	Conclusions and Future Directions	51

Robotic Grasping and Manipulation

Antonio Bicchi and Vijay Kumar		55
1	Introduction.....	55
2	Kinematics of Manipulation	57
3	Grasp Closure Properties	60
3.1	Form Closure.....	60
3.2	Force Closure.....	61
4	Dynamics	63
5	Stability	64
6	Contact Compliance.....	66
7	Grasping and the Kinematics of the Hand	67
8	Measures of Grasp Performance.....	67
9	Concluding Remarks	68

Tactile Sensing for Robotic Manipulation

Claudio Melchiorri		75
1	Introduction.....	75
2	An Overview on Force/Torque and Tactile Sensors	77
2.1	Force Sensors, IT Sensing	77
2.2	Tactile Sensors	79
2.3	Technology for Tactile Sensing	80
3	Examples of Force/Torque and Tactile Sensors.....	82
3.1	A Miniaturised Intrinsic Tactile Sensor	83
3.2	A Tactile Matrix Sensor	84
4	Slip Detection	87
4.1	Slip Detection with a Tactile Sensor	87
4.2	Slip Detection with a Tactile and Force/Torque Sensor.....	89
4.3	Rotational and Translational Slip Detection	90
5	Experimental Results.....	93
5.1	Translational Slip Detection and Control.....	94
5.2	Rotational and Translational Slip Detection and Control	94
5.3	Manipulation Exploiting Friction.....	96
6	Conclusions	99

On Control of Flexible Robots

Antonio Ficola, Mario Luca Fravolini, and Michele La Cava	103
1 Introduction	103
2 Modelling	104
2.1 Single Link	104
2.2 Exact Model	104
2.3 Rayleigh-Ritz Methods	106
2.4 Finite Element Method	107
2.5 Multiple Links	107
2.6 Remarks	108
3 Control	109
3.1 PD Regulation	109
3.2 Strain Feedback Control	109
3.3 Inverse Dynamics	109
3.4 Fuzzy Regulation	110
3.5 Robust Control	110
3.6 Singular Perturbation	111
3.7 Sliding	111
3.8 Input Shaping	112
3.9 Cyclic Control	112
3.10 Predictive Control	112
4 Conclusion	117

Interaction Control

Fabrizio Caccavale, Ciro Natale, Bruno Siciliano, and Luigi Villani	121
1 Introduction	121
2 Motion Control	123
2.1 Modelling	123
2.2 Motion Control in Task Space	124
3 Six-DOF Impedance Control	126
3.1 Classical Impedance Control	127
3.2 Geometrically Consistent Impedance Control	129
3.3 Inner Motion Control Loop	132
3.4 Redundancy Resolution	133
4 Cooperative Robots	135
4.1 Loose Cooperation	135
4.2 Tight Cooperation	136
5 Experimental Validation	138
5.1 Single Manipulator	139
5.2 Loose Cooperation	148
5.3 Tight Cooperation	150
6 Conclusion and Future Directions	152

Impact Modelling and Control of Robotic Links

Marina Indri	155
1 Introduction.....	155
2 Impact Modelling	157
2.1 Some “Pioneering” Works	157
2.2 General Approaches to Impact Modelling	158
3 Impact Control in Robotics	161
3.1 Some “Old” Experimental Switching Schemes	162
3.2 Integral Control Laws and Bouncing Problems	163
3.3 A Unique Control Law for Free Motion and Contact	165
4 Some Proposed Impact Control Schemes	165
4.1 A Control Scheme Based on Impact Effects Estimation	167
4.2 A Position Feedback Control Law	171
5 Conclusions	177

Control of Wheeled Mobile Robots: An Experimental Overview

Alessandro De Luca, Giuseppe Oriolo, and Marilena Vendittelli	181
1 Introduction.....	181
1.1 Organization of Contents	182
2 Basic Motion Tasks	183
3 Modelling and Control Properties	184
3.1 Controllability at a Point	185
3.2 Controllability About a Trajectory	186
3.3 Feedback Linearisability	187
3.4 Chained Forms	188
4 Target Vehicle: SuperMARIO.....	189
4.1 Physical Description.....	189
4.2 Control System Architecture	190
4.3 Kinematics	191
4.4 Control Constraints	192
5 Trajectory Tracking	193
5.1 Feedforward Command Generation	193
5.2 Linear Control Design	194
5.3 Nonlinear Control Design	195
5.4 Dynamic Feedback Linearisation	196
5.5 Experiments.....	199
6 Posture Stabilisation	204
6.1 Smooth Time-varying Control	204
6.2 Nonsmooth Time-varying Control	207
6.3 Control Based on Polar Coordinates	210
6.4 Dynamic Feedback Linearisation	215

7	Guidelines for End-users	221
7.1	Summary and Comparison	221
7.2	Future Directions	223

Landmark Recognition in Indoor Navigation by Fuzzy Maps and CBR

Alessandro Micarelli, Stefano Panzieri, Lorenzo Sciavicco, and Giovanni Ulivi		227
1	Introduction	227
2	World Representation	228
3	The Overall Structure	230
4	Fuzzy Maps	231
5	Fuzzy Morphology	233
6	Worldmark	233
7	Wavelet Decomposition	235
7.1	Denoising with Wavelet	236
8	Similarity Metric	238
8.1	Cross Correlation Similarity	238
8.2	Fuzzy Similarity	238
8.3	Wavelet Similarity	240
9	Extension to a Dynamic Library: CBR	240
10	Combined Recognition	244
11	Experiments with Static Library and TBM	246
12	Conclusion and Future Developments	249

Sensor Fusion for Robot Localisation

Fabrizio Grandoni, Agostino Martinelli, Francesco Martinelli, Salvatore Nicosia, and Paolo Valigi		251
1	Introduction	251
2	Sensor Integration	254
2.1	Techniques and Approaches	254
2.2	Outdoor and Indoor Characterisation	257
3	Sensor Fusion	257
3.1	Sensor Fusion With Known Statistics in a Centralised Architecture	258
3.2	Sensor Fusion With Known Statistics in a Decentralised Architecture	260
3.3	Sensor Fusion With Unknown Statistics	261
4	Localisation	262
4.1	The Localisation Problem	262
4.2	The Extended Kalman Filter	264
4.3	Adaptive Extended Kalman Filter	266
4.4	Integrating Vision and Sonar	267
4.5	Observability and Localisation	267

4.6	Simultaneous Localisation and Map Construction	267
4.7	Multi-robot Localisation	268
4.8	Some Experimental Results	268
5	Conclusion and Future Directions	269

Strategies for Control and Coordination within Multiarm Systems

Giuseppe Casalino, Damiano Angeletti, Tommaso Bozzo, and
Giorgio Cannata

Dipartimento di Informatica Sistemistica e Telematica
Università degli Studi di Genova
<http://www.graal.dist.unige.it>

Many of the aspects related with the definition and development of a functional and algorithmic architecture for multiarm systems are dealt with in this chapter. Emphasis is given to the role played by each layer and to their composing basic functional blocks. In particular it is shown how such basic blocks can be effectively used to the aim of incrementally constructing complex cooperative tasks, via suitable and simple interconnecting rules. An example of Hw/Sw architecture supporting the entire functional and algorithmic architecture is also provided.

1 Introduction

The problem of controlling and coordinating the actions of multiarm systems capable of performing complex cooperative operations, such as for instance coordinated object handling and transportation, coordinated assembly, etc., may result in a formidable task if not approached in a structured and well ordered way. To this end, any well structured approach to the problem should actually involve three different aspects, which are however very correlated among them; namely: a) the functional and algorithmic architecture of the overall coordination and control system, b) the possibly distributed Hw/Sw architecture supporting the functional and algorithmic one, c) the efficiency and quality of the tools and environments used for developing the corresponding real time Sw.

Moreover, as clearly stated in [15], the “demand for an organisation or structure that relies on well defined concepts to enable the effective realisation of complex system” is clearly recognised for each one the cited items. As a matter of fact, research in these interrelated areas of interests for robotics has progressively increased, in parallel with the theoretical and technical progresses achieved in robot control, real-time operating systems and software engineering as well and, certainly, it has also been driven by the increasing progresses in computer technology.

As a matter of fact, all a set of very challenging experiments, up to now performed (like for instance the Rotex arm experiment in space, the Sojourner sytem exploring Mars, or also the P2 and now Asimo walking humanoids, just

to mention some among the most impressive ones) have clearly brought into evidence the effectiveness of the results and performance that can be obtained whenever complexity is approached in a well structured and organised way.

A very good survey and collection of selected papers, covering each one of the mentioned areas, is represented by [1], where the above items are presented from both the conceptual and implementative point of view, and where a set of general robotic examples are also given, together with indication relevant to the real-time Os and robotic oriented Sw developing tools available at that time (see for instance [5], [6], [19] as representative papers dealing with the above three items respectively).

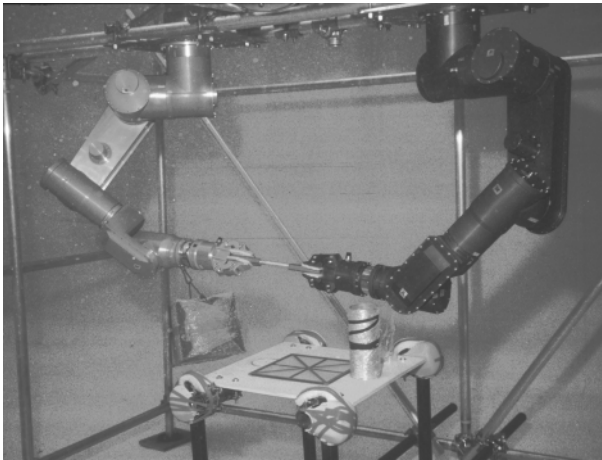


Figure 1.1. Underwater test bed

Naturally enough, as it concerns this last item, further improvements and also new products actually emerged during the successive years, even without introducing substantial changes in overall approach philosophy accessed at that time. Among the three previously mentioned items, and possibly due to the peculiarities exhibited by the robotic field, a key role is however played by the first one (i.e. functional and algorithmic architectures) which substantially drove the robotic oriented part of most of the proposed developing environment.

Within this chapter, many of the aspects related with the development of a functional and algorithmic architecture for multiarm systems is addressed in some details. Special emphasis is given to the structure of the various layers composing the architecture and, more in particular, to the different building blocks and the mechanisms for their composition, within the execution of complex cooperative tasks. To this end, and without loss of generality, reference will be made to the dual arm working cell reported in Figure 1.1,

composed by two 7-DOF electrically actuated arms, and employed for underwater applications [14], [13], [10], [11]. Moreover, just as an illustrative example, some little details concerning the adopted distributed Hw/Sw architecture implementing the overall functional one will also be provided, together with some general indications about the employed developing tools and environments.

2 The Functional Architecture

The overall functional architecture adopted for the command and control of multiarm systems corresponds to the one reported in Figure 2.1, here specialised for the considered case of two arms.

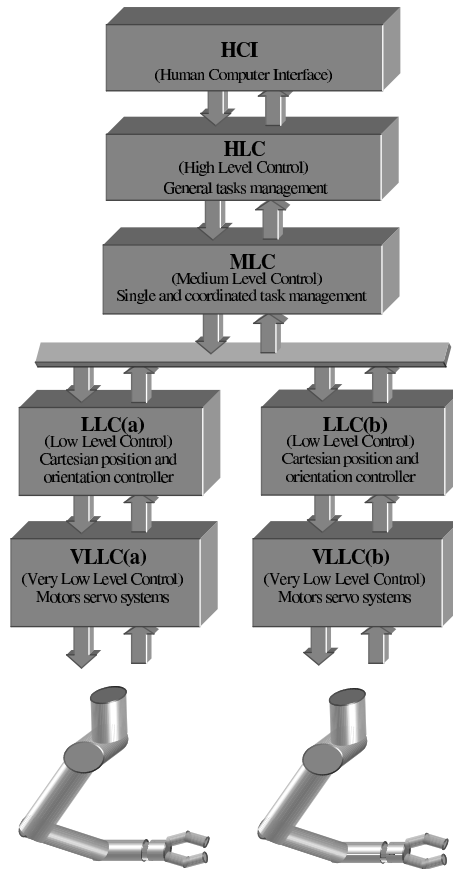


Figure 2.1. Functional architecture for the dual arm workcell

As many others robotic functional architectures, it also is organised as a three/four-level hierarchical structure, where each level reflects a layer of abstraction of the solution of the global command and control problem. As a consequence, this results in an efficient yet flexible framework for the organisation of the interaction of relevant hardware/software components. To this end the description which follows will also clarify the rationale underlying the specifically proposed multilayered structure.

By analysing the hierarchy starting from the bottom, we first find the so-called “Very Low Level Control” (VLLC) layer, in correspondence of which joint velocity control is independently performed for each one of the arms, being the relevant reference signals provided by the next upperlying level. Typically, in practical applications (as it actually is, for the one referenced here) the VLLC is directly provided by the power drives of the joint motors, each one embedding a simple “high gain” PI velocity regulator. In alternative, more sophisticated controllers, like PI regulators plus exact or approximate, or even adaptive, nonlinearity compensation methods, could also be employed at this level. In both cases, however, the net result associated with the control laws implemented at this level is that of reducing the behaviour of each one of the subsystems “VLLC + Arm” similar to that of a bank of pure integrators, which directly transform (via time integration) the reference joint velocities into their corresponding joint positions.

By proceeding upward, the next layer of the hierarchy is represented by the so-called “Low level Control” (LLC) accomplishing the function of independently controlling the Cartesian position and orientation (or even imposing the angular and linear velocities) of the tool frame of each one of the manipulators. More precisely, inside each one of the blocks composing such layer, the Cartesian position and attitude of the tool frame of the associated manipulator are real-time evaluated on the basis of joint position measurements and knowledge of the corresponding robot geometry. The output of such evaluation represents the Cartesian feedback information which is compared, inside each LLC block, with the corresponding Cartesian command provided by the next upperlying level. The further processing performed inside each one of the LLC blocks finally produces, as outputs, the reference joint velocity signals to be input to the corresponding VLLC block.

Still proceeding upward along the hierarchy we then find the so-called “Medium Level Control” (MLC) layer, whose function is that of suitably coordinating the motions of the manipulators tool frames in correspondence of any activated task. This is done via the automatic generation of the most suitable Cartesian reference positions and orientations to be tracked by the robots tool frames. Obviously enough, inputs to the MLC layer are the current positions and attitudes of both tool frames (provided as feedback by the underlying LLC layer) plus the commands specifying the tasks to be activated.

At the next level we further find the so-called “High Level Control” (HLC) layer, implementing the methods for the automatic composition (sequential and/or parallel) of tasks into more complex ones (macro-tasks) plus the relevant scheduling techniques. The outputs from HLC are consequently the task commands for the underlying MLC block. Inputs to HLC are instead those provided as feedback by MLC, which inform about the current state of execution of the activated tasks, plus the request commands for new macro-tasks to be successively performed.

Finally, the hierarchy ends up with the upmost level, simply represented by the “Human Computer Interface” (HCI), hosting the whole interactive Sw needed to the operator for efficiently commanding and monitoring the entire system.

As it should be now apparent, the proposed functional architecture clearly results in a hierarchy organised in such a way as to maintain its lowest layers VLLC and LLC (the first operating at joint level and the second at the “interface” between joint and Cartesian space) within an algorithmic and processing structure which is always independent, and consequently invariant, with respect to the variety of tasks to be executed; being the variability with the tasks of the associated control schemes strictly confined within the upper MLC and HLC layers of the architecture.

As it will be better clarified in later sections, such variability of the control schemes, at the MLC layer, can however be efficiently handled by simply enabling, in correspondence of each task to be activated, the right interconnections among the elements of a restricted set of pre-defined, invariant, elementary building blocks. Obviously enough, the function of structuring such interconnections among the basic building blocks is instead assigned to the HCL layer. Moreover note how such upper layers MLC and HLC, since totally operating at a Cartesian level, actually result to be largely independent from the specific geometry and kinematic of the underlying mechanical structure (here “only occasionally” represented by a couple of 7-DOF manipulator arms).

Before concluding the present section, finally observe that, when needed, even the so-called “Motion and Interaction Control” (i.e. the simultaneous control of both motion and reaction forces/torques arising during the interaction with the environment and/or manipulated objects) can also be performed at the MLC layer of the architecture (thus extending its operative space from geometric-Cartesian only, to geometric-Cartesian + force/torque-Cartesian) provided that suitable force/torque sensors feedback information is made available at this level. In this case, however, since the underlying LLC blocks can only receive position/orientation or linear/angular velocity commands, the MLC must consequently perform the force/torque interaction control via the generation of “equivalent” Cartesian position and/or velocity reference signals, to be distributed to the underlying LLC blocks. This will be better illustrated within a specifically dedicated later section of this chapter.

From now on, the present chapter will proceed with an illustrative example of a possible Hw/Sw architecture supporting the above presented functional one, while the remaining sections will be dedicated to a more detailed description of each layer and related, composing, functional elements.

3 An Example of Supporting Hw and Sw Architecture

As it has been already mentioned, the definition of a functional architecture also provides a framework for the organisation of the interaction of relevant Hw/Sw components. In this section, an illustrative example of this fact is briefly presented with respect to the referenced dual arm workcell of Figure 1.1.

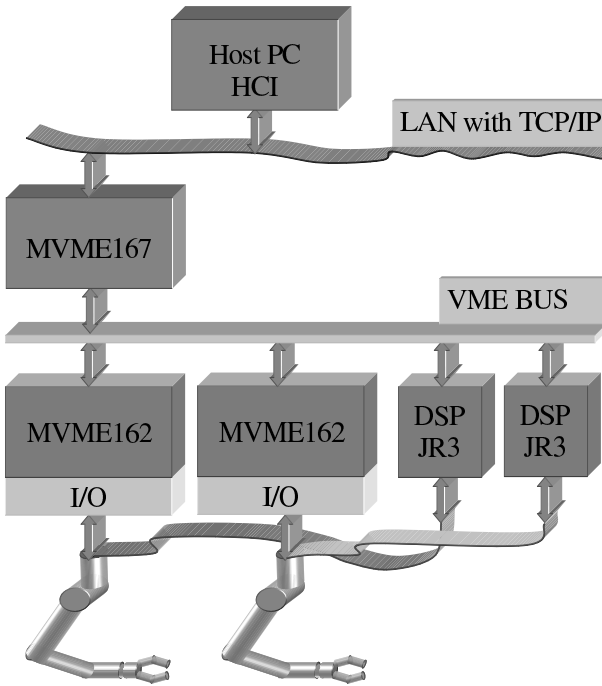


Figure 3.1. Hardware architecture description

The adopted Hw architecture corresponds to the one indicate in Figure 3.1, where a standard distributed commercial architecture, based on VME bus and Motorola 68040 processing units, have been used. In this framework the two MVME 162 boards (one for each arm) host the LLC layer, while the

single board MVME 167 is used for hosting both the MLC and HLC layer. (The VLLC layer is instead embedded within the power drives, not shown in the figure, directly acting on the motor joints). The HCI is instead hosted by a PC interacting with the MVME 167 via LAN connection.

The two MVME 162 boards directly acquire the encoder signals provided by the power drives, via local encoder interfaces installed on the boards, and provide analog voltage commands to the power drives via D/A converters, they also directly installed on the same boards.

All the remaining data exchange among the three boards occur on the VME bus, on the basis of sharing memory communication techniques. The same kind of data exchange is also established between the MVME 167 board and the two DSP based force/torque sensor acquisition units, each one associated with the relevant wrist force/torque sensor of the corresponding arm.

Communications between the HCI hosting PC and the entire system are instead established by the LAN connection with the MVME 167 board, via the use of TCP/IP protocols (sockets). Computing board MVME 167 also acts as “master board”; thus meaning that all the synchronisation aspects concerning data exchanges and communications are handled, within the architecture, in a centralised way.

The overall Hw architecture is driven under the supervision and control of the real-time distributed VxWorks (of Windriver) operating systems. A common and synchronised sampling time of 5 ms could be chosen for all the control processes running on the different computing boards; this apart the processes regarding the regularised pseudo-inversion of some Jacobian matrices, performed via SVD algorithms, whose sampling time had to be fixed to 30 ms due to their greater computational burden (thus meaning that the upgrading of a Jacobian matrix always occurs every five sampling intervals of 5 ms, which however turned out to be acceptable in terms of obtained performances).

Naturally enough, even other types of computing boards, buses, and real-time OS (like for instance Qnx or RT-Linux, just to mention some of the more mature ones by now) could also be used, without however changing the underlying, functionally driven, Hw/Sw design philosophy.

At this point, in order to conclude the example, a brief mention of the used design and developing tools could also be useful. More specifically, the development of the real time supporting Sw architecture has been done on the basis of the real-time Sw developing and debugging environment Tornado (of Windriver), the development of the C-code corresponding to the various functional blocks (see forthcoming sections) has been done via the Matlab/Simulink/RT-Workshop (of The Mathworks) environment (simulation, automatic C-coding and downloading), while the C-coding of specific robotic functional blocks, such as transformation and Jacobian matrices, etc., has been obtained automatically by exploiting the facilities offered by the so-

called RDS (Robotic Developer Studio) [11], [17]: a proprietary environment devoted to symbolic modelling and simulation of complex robotic systems.

4 The Low Level Control Layer

Consider Figure 4.1, where the kinematic structure of one of the 7-DOF arms of the workcell is schematically represented. In this figure, T_e is the transformation matrix of the end-effector frame $\langle e \rangle$ w.r.t. base frame $\langle o \rangle$, T is the (constant) transformation matrix of the tool center frame $\langle t \rangle$ w.r.t. $\langle e \rangle$, and T^* the (generally time varying) transformation matrix of the reference frame $\langle g \rangle$ (w.r.t. $\langle o \rangle$) to be tracked by tool frame $\langle t \rangle$. Moreover, vectors r and ρ (both projected on the base frame $\langle o \rangle$) represent the distance and the misalignment (equivalent rotation vector) of reference frame $\langle g \rangle$ w.r.t. $\langle t \rangle$. By collecting together the two error vectors, the six dimensional global error vector e is defined as $e = [r^T, \rho^T]^T$.

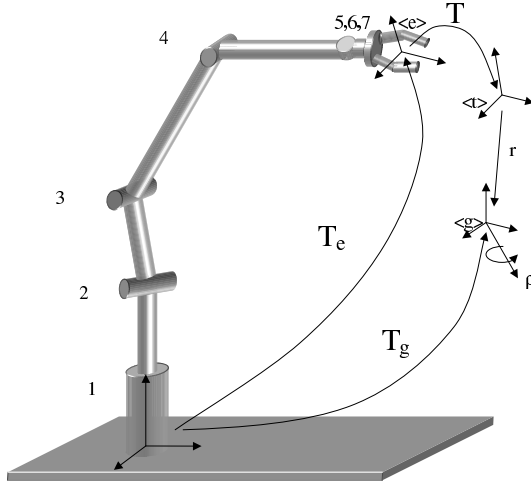


Figure 4.1. Schematic representation of a single arm

The control scheme for a single arm is schematically depicted in Figure 4.2, where the block “Robot + VLLC” represents the physical arm equipped with its seven joint drivers, each one implementing a closed loop velocity control at the corresponding joint. Thus such block can be seen as a compact one receiving reference joint velocities as input, while giving the corresponding joint positions as their time integral output.

The remaining part of the scheme, termed as the “Low Level Control” (LLC) loop, is instead composed by the interconnection of the following

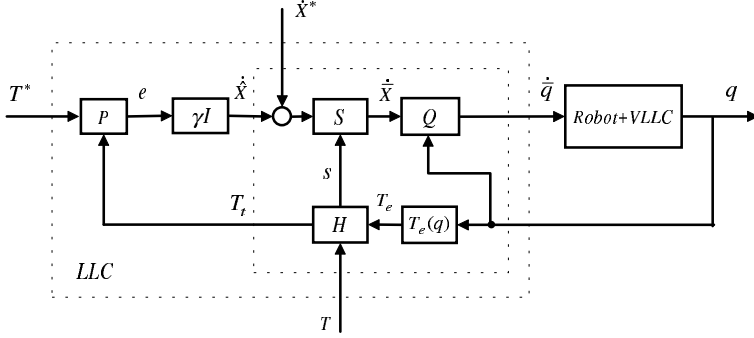


Figure 4.2. Single arm control scheme

blocks. The processing block P, used for real time evaluating the global error e via the solution of the well known “versor lemma” equations [12], [4], for its rotational part ρ , and via the difference between the first three elements of the last columns of T^* and T_t , respectively, for its linear part r . The gain matrix gI ($g > 0$), used for translating the global error e into the generalised Cartesian velocity vector $\dot{\hat{X}} = [\hat{\omega}^T \ \hat{v}^T]^T$ (also projected on $\langle o \rangle$) to be assigned to the tool frame T_t in order to have the convergence of e toward zero. Then we have the additional Cartesian velocity input \dot{X}^* , only used for coordination purposes (as described in Section 4), and moreover block S, translating $\dot{\hat{X}}$ into the related Cartesian velocity $\dot{\hat{X}}$ for the end-effector frame $\langle e \rangle$, via the use of the well known rigid body velocity relationships

$$\begin{cases} \bar{\omega} = \hat{\omega} \\ \bar{v} = \hat{v} + [s \wedge] \hat{\omega} \end{cases} \quad (4.1)$$

where s is the vector distance (projected on $\langle o \rangle$) of frame $\langle t \rangle$ with respect to $\langle e \rangle$; i.e. vector

$$s = R_e t \quad (4.2)$$

being R_e the rotation matrix part of the end-effector frame transformation matrix T_e , and t the same (constant) distance vector projected on the end-effector frame $\langle e \rangle$ (i.e. the first three elements of the last column of the constant transformation matrix T).

Finally, within the same scheme, block H is used for implementing the simple transformation relationship

$$T_t = T_e T. \quad (4.3)$$

The end-effector Cartesian velocity reference signal $\dot{\hat{X}}$, generated as indicated above, is in turn transformed, via the functional block Q, into a corresponding set of joint velocity reference inputs, globally represented by the seven dimensional vector $\dot{\hat{q}}$.

In Figure 4.1, the sub-scheme enclosed in the inner dotted box constitutes what is generally termed as the “Interface between the Cartesian and the joint spaces” of the overall scheme. As a matter of fact, within such interface part block Q is also required to cope with aspects that generally go a little beyond the simple pseudo-inversion of the Jacobian matrix.

Just as an illustrative example, in the following we briefly recall the nature of such aspects by showing how they have been handled within the specific case of the considered workcell.

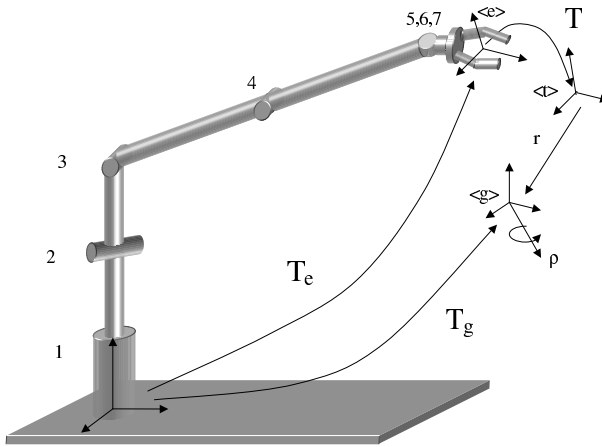


Figure 4.3. Singular configuration

To this aim consider again the redundant arm structure as depicted Figure 4.3, showing the most important singular configuration exhibited by the arm. As it is well known, in the vicinity of such configuration, joint velocities produced via simple pseudo-inversion of the Jacobian matrix would tend to infinite values; thus inducing (at least temporarily) instabilities and not acceptable vibrations within the whole structure. Thus the need for regularisation naturally arises, together with that of maintaining the arm sufficiently “far” from such configuration though. Then, in order to satisfy the above requirements, the following algorithm has been finally adopted for generating $\ddot{\mathbf{q}}$ from the inside of the interface block Q

$$\dot{\tilde{q}} = W^{-1}J^T(JW^{-1}J^T + \lambda I)^{-1}(\dot{\tilde{X}} - h_4\dot{\tilde{q}}_4) + \dot{\tilde{q}} \quad (4.4)$$

being

$$\begin{aligned}\ddot{\vec{q}} &= [0 \quad 0 \quad 0 \quad \ddot{\vec{q}}_4 \quad 0 \quad 0 \quad 0]^T \\ \dot{\vec{q}}_4 &= -k(q_4 - \bar{q}_4).\end{aligned}\tag{4.5}$$

The rationale underlying the above expression can be simply explained as follows. First of all note that term \ddot{q} corresponds to an elastic feedback tending to maintain the fourth joint far from its zero value. The relevant elastic gain has been chosen in such a way as to substantially act only in the vicinity of the zero position, thus resulting in a finite support, bell shaped, positive radial basis function of q_4 centered on zero. Due to the presence of \ddot{q} , the first term in (4.4) is consequently used for both compensating its Cartesian effects $h_4\ddot{q}$ (h_4 is the fourth column of the end-effector Jacobian matrix J) while also guaranteeing the tracking of the required Cartesian velocity \dot{X} . This obviously corresponds to perform a projection of \ddot{q} on the null space of J , which can however be done by minimally depressing the elastic action given by \ddot{q}_4 , via the use of the following Jacobian weighting matrix W

$$W \doteq \text{diag}[1, 1, 1, w, 1, 1, 1] \quad w \doteq 1 + k. \quad (4.6)$$

Finally note in (4.4) how the term λ is also added for regularisation purposes, taking also the form of a finite support, bell shaped, positive radial basis function (centered on zero) of the determinant of the matrix $JW^{-1}J^T$ (or, even better, of the ratio between its minimum and maximum singular value) [18]. With the adoption of the above technique, it can be shown that singularity can always be avoided provided that \dot{X} is guaranteed to always lie within specific norm bounds, which can however be enlarged for increasing values of k [12].

The structure of the overall scheme of Figure 4.1 has been proposed by some of the authors since 1995 [2]. Its nice stability and robustness properties, with respect to possible uncertainties of both dynamic and geometric nature, has also been proved by the authors within different reports and papers. Theoretical details can be found for instance in the works [2], [7], [12], [4], [3].

5 The Medium Level Control Layer

As discussed in Section 2, the major function to be accomplished by the MLC layer is that of guaranteeing motion coordination during the execution of a set of tasks generally involving both the arms; being those requiring the motion of a single arm only considered as a very special case of the wider class of cooperative ones. Nevertheless, since the capability of efficiently structuring the motion of a single arm, actually constitutes the fundamental framework for similarly structuring more complex cooperative tasks, the present section will consequently start with a discussion about single arm motion tasks, while a treatment of the more general cooperative ones will follow immediately after.

5.1 The Teleoperated/Automatic Independent Motion Control of the Arms

A “Teleoperated Motion Control Mode” of a single arm is intended to be established when the reference frame to be tracked by each arm is made to move around space by integrating velocity data (angular and linear) acquired from HCI, via the use of suitable “space mouse” devices interacting with the HCI itself. However, as it will be better explained shortly, a simple integration of the velocity data coming from the space mouse devices may actually not be effective for driving each one of the arms.

As a matter of fact, the more complex functional diagram depicted in Figure 5.1 has revealed to be the most suitable one for teleoperation purposes [12]. In order to explain the rationale underlying the use of the proposed scheme, let us first assume for a while that blocks named gI and $\bar{g}I$ simply correspond to the identity and the null one, respectively (i.e. $g = 1$ and $\bar{g} = 0$); then in this case the diagram directly reduces to the sole integration, performed by the block named Int , of the acquired absolute velocity data ω_g and v_g assigned to the goal reference frame $\langle g \rangle$.

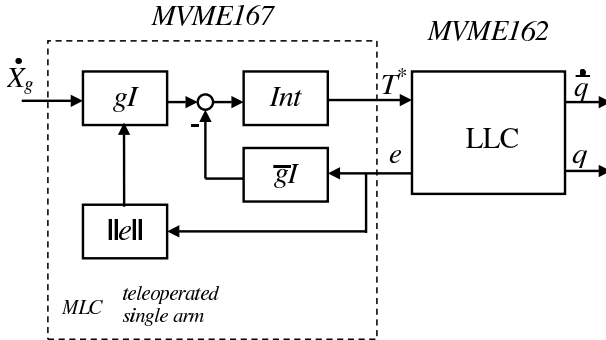


Figure 5.1. Functional diagram for teleoperation purposes

Obviously enough, provided that the absolute velocities data ω_g and v_g are interpreted as projected on the absolute frame $\langle o \rangle$, block Int results in performing the time integration of the following differential equations (with the first of the two efficiently integrated via the use of the well known “Rodriguez formula” [16])

$$\begin{cases} \dot{R}^* = [\omega_g \wedge] R^* \\ \dot{p}^* = v_g. \end{cases} \quad (5.1)$$

In the above equations the starting conditions are set to be equal to the initial posture assumed by the tool frame $\langle t \rangle$ of the arm. The resulting absolute

reference outputs R^* and p^* (rotation matrix and position of $\langle g \rangle$ w.r.t. $\langle o \rangle$ projected on $\langle o \rangle$) are then assembled together in order to give rise to the corresponding transformation matrix T^* . It should be however noted that in this case, provided that the sole block Int is actually used, there might be the risk of having (at least temporarily) the tracking error e attaining very high values, before eventually converging to zero.

This might for instance happen in case of a “fast” virtual motion of the space mouse or, more commonly, when the target frame $\langle g \rangle$ is pulled too far outside the robot workspace. In particular, since in this latter case the robot is progressively pushed toward its singularity, while trying to reduce errors that instead (due to mechanical constraints) could even attain unbounded values, very high values for \dot{X} could consequently be induced within the scheme of Figure 4.2. Then, just due to this, it turns out that the assumed norm bounds of \dot{X} (i.e. those allowing the effective avoidance of singularity, as mentioned in Section 4) generally risk to be seriously exceeded.

As a matter of fact, since this drawback could be avoided by simply forcing the tracking error e to remain confined within a specified superior norm bound, this is consequently done via the use of the block gI , having the gain g taking on the form of a finite support, bell shaped, positive radial basis function centered on zero, of the tracking error norm. As it can be easily seen, since the velocities of the target frame are now progressively reduced toward zero in case the error e approaches the specified norm limit, it consequently occurs that the error e itself cannot ever overpass such norm bound (obviously provided that it always starts from lower norm values).

Nevertheless, with the sole use of the above described “norm error limiter” block it is clear that, in case of attainment of the specified norm error bound in correspondence of points located outside of the robot workspace (where the error itself cannot be anymore reduced via robot motion, and where the velocity gain g however attains the zero value) unfortunately there will not be the possibility of moving again the goal frame $\langle g \rangle$: this due to the fact that, in the impossibility of reducing the error, its velocity will be consequently maintained at a null value by the gI block itself. Then, in order to avoid such additional drawback, a moderate, tracking error based, constant velocity feedback has been added via the constant gain block termed as $\bar{g}I$. As a matter of fact, since such block always exerts a moderate recalling action on $\langle g \rangle$ itself, directed toward the tool frame $\langle t \rangle$ of the arm, it always prevents the error e to exactly reach its upper norm bound; thus always avoiding the total zeroing of the gain in the gI block.

Obviously enough, when the arm is instead inside its workspace, the motion capabilities of the arm itself (i.e. the capability guaranteed by the LLC gains of maintaining e inside its lower limits closed to zero, provided e itself starts from the inside of such limits) simply result in a “masking” of the (moderate) feedback action produced by the $\bar{g}I$ block. The final assessment

of the functional scheme reported in Figure 5.1 corresponds to what has been termed as the “MLC Teleoperated single arm”.

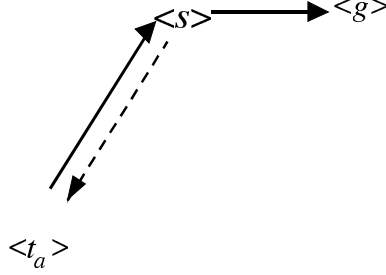


Figure 5.2. Frame tracking arrangement for the automatic independent motion

As opposed to the above described teleoperated mode, an “Automatic Independent Motion Control Mode” for a single arm is intended to be established when the end effector of one of the arms is required to reach any assigned position/orientation in space, starting from any initial posture.

Within such kind of control (reentering within the more general category of the so-called “point-to-point” control modes) the position/orientation trajectory followed while reaching the final posture is automatically established by the implemented corresponding control loop. Then, it is just in this sense that such kind of control mode differs from the already considered teleoperated one, where the trajectory is implicitly assigned on line by the operator interacting with the system (via a reference velocity profile, generated by a space mouse device).

In order to implement such kind of automatic control mode it should be however clear that, at least in principle, the sole LLC fundamental scheme of (Figure 4.2) would actually suffice, provided that it could be somehow assured that the induced internal Cartesian velocities remain norm bounded, in order to allow singularity avoidance (see Section 4).

As a matter of fact, similarly to the teleoperated case, since a way for guaranteeing such possibility is still that of maintaining the global error e within a suitably assigned maximum norm bound, the conceptual scheme of Figure 5.2 can consequently be adopted.

In such scheme, frame $\langle s \rangle$ is an auxiliary one that, starting from a posture coinciding with the tool frame $\langle t \rangle$, is commanded to reach the goal frame $\langle g \rangle$ via the use of a Cartesian control loop whose feedback gain takes on the same form of a finite support, bell shaped, positive radial basis function centered on zero, of the norm of the error e between frame $\langle s \rangle$ itself and $\langle t \rangle$.

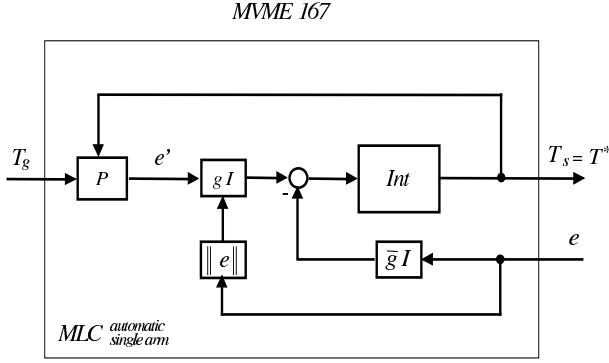


Figure 5.3. Medium Level Control for automatic independent motion

Meanwhile frame $\langle t \rangle$ is commanded to follow $\langle s \rangle$ (and a moderate constant feedback action of the form $\bar{g}I$ still exists, weakly recalling $\langle s \rangle$ toward $\langle t \rangle$). As it can be easily realised, this guarantees the boundedness of the tracking error e (of $\langle s \rangle$ w.r.t. $\langle t \rangle$), and also its convergence to zero together with the other tracking error e' (of $\langle g \rangle$ w.r.t. $\langle s \rangle$) provided $\langle g \rangle$ is inside the arm workspace. Otherwise ($\langle g \rangle$ outside the arm workspace) both the errors e, e' will only remain bounded (in particular with the norm of e inside its upper limit) eventually reaching the equilibrium condition

$$g(\|e\|^2)e' - \bar{g}e = 0 \quad (5.2)$$

only satisfied by a corresponding vector e whose norm is necessarily strictly lower than its assigned upper bound.

The functional scheme corresponding to the above conceptual one is reported in Figure 5.3.

As it can be seen from the figure, the part of control concerning tracking of $\langle s \rangle$ by part of $\langle t \rangle$ is obviously let to the LLC of the arm. Only the remaining part of the scheme is implemented as MLC.

5.2 The Teleoperated/Automatic Coordinated Motion Control of the Arms

Consider the situation depicted in Figure 5.4 showing the (assumed steady) condition when a lightweight rigid object has been firmly grasped by both the grippers, after a preliminary phase of object approach and consequent grasping, performed via independent control mode. Then the same figure clearly evidences the need for being able to move both the end effectors without varying their mutual position and orientation; thus letting the grasped object consequently move properly. To this end also assume that, prior to any

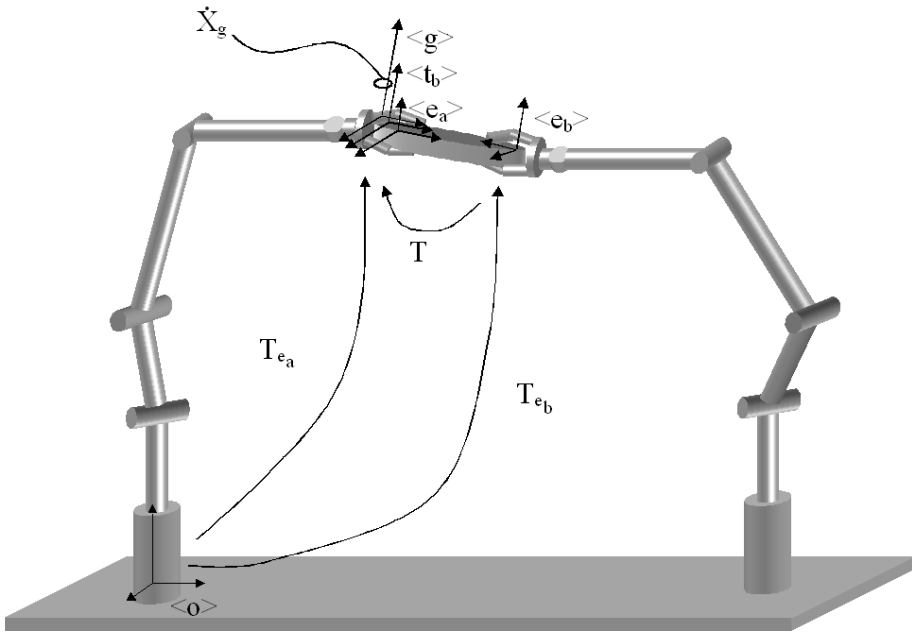


Figure 5.4. Teleoperated dual arm workcell handling an object

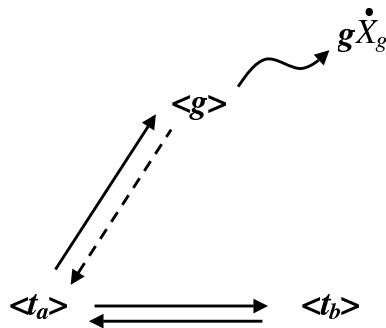


Figure 5.5. Frame tracking arrangement for teleoperated coordinated motion

motion, the existing transformation matrix T between the two end effectors is preliminary and automatically evaluated (on the basis of the known robot geometry information) via the simple expression

$$T = T_{e_b} T_{e_a}. \quad (5.3)$$

Then a tool center frame $\langle t_b \rangle$, having the constraint matrix T as transformation matrix w.r.t. $\langle e_b \rangle$, is assigned to arm (b) while the tool center frame $\langle t_a \rangle$ of arm (a) is let to coincide with its end-effector frame $\langle e_a \rangle$.

The teleoperated coordinated motion of the arms can be performed on the basis of a “parallel task composition”, to be implemented in the following way. 1) The tool frame $\langle t_b \rangle$ is required to remain coinciding with the tool frame $\langle t_a \rangle$. 2) The tool frame $\langle t_b \rangle$ is required remain coinciding with the tool frame $\langle t_b \rangle$. 3) Meanwhile, the tool frame $\langle t_a \rangle$ is required to closely follow the moving goal frame $\langle g \rangle$ (supposed to be initially coinciding with $\langle t_a \rangle$). Such situation, actually indicated in Figure 5.4, conceptually corresponds to what is reported in Figure 5.5, where the dotted arrow is used for indicating the “attractive” action contemporaneously performed on $\langle g \rangle$ toward $\langle t_a \rangle$, by the (moderate) feedback gain $\bar{g}I$, still accomplishing the same role as that described in Section 3.

As it is apparent from Figure 5.5 (and in accordance to what has been above established) while frame $\langle t_b \rangle$ is required to track a single frame (actually frame $\langle t_a \rangle$) and frame $\langle t_a \rangle$ is instead required to contemporaneously follow two different target frames (namely, frame $\langle t_b \rangle$ and the moving one $\langle g \rangle$). The same obviously holds also for frame $\langle g \rangle$ itself, which is required to move around space while simultaneously tracking, due to the presence of the previously mentioned moderate feedback action, also frame $\langle t_a \rangle$. To this regard, it must be noted that such seemingly contrasting aspects actually are only of apparent nature: this simple due to the fact that the above established global task actually admits a unique solution, contemporaneously fulfilling to all the composing subtasks (i.e. the one corresponding to have, eventually or just from the beginning, all the frames coinciding among them; see [8]).

Seen from the MLC implementative point of view, the control scheme allowing the execution of the above specified overall task consequently results into that of Figure 5.6, corresponding to a suitable composition of the same control modules already defined within the independent control of the arms.

Within this figure, the two blocks LLC (a) and (b) located at the right are the same control schemes, already used for the independent control of the two arms; with the difference that, now, each LLC layer block receives, as input reference frame to be tracked, the tool frame of the other. Moreover block LLC (a) also receives, as an additional input, the Cartesian velocity control signal \dot{X}_a^* generated by the external loop (the one enclosed in the inner dotted box) delegated to control the tracking of $\langle g \rangle$ by part of $\langle t_a \rangle$. Further, the transformation matrix T^* of the moving frame $\langle g \rangle$

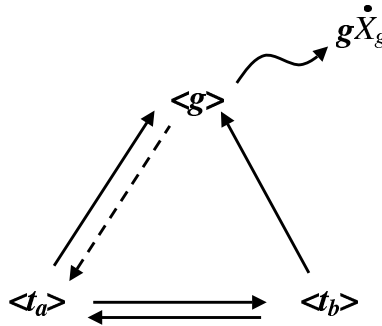


Figure 5.7. Another possibility of frame tracking arrangement for teleoperated coordinated motion

is also required to follow the moving frame $\langle g \rangle$, as it is for $\langle t_a \rangle$ within the diagram of Figure 5.5.

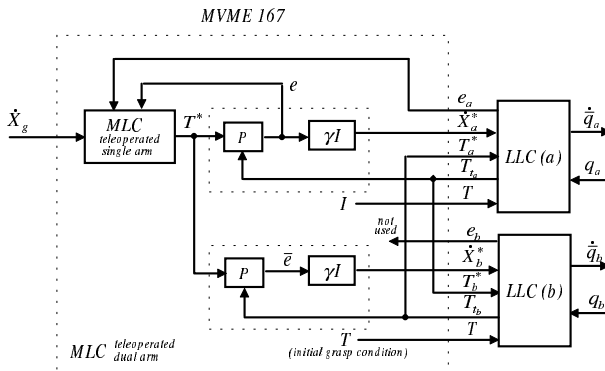


Figure 5.8. Control scheme for teleoperated coordinated motion, corresponding to the conceptual scheme of Figure 5.7

As a result of such modification, the inherent priority previously existing between the tasks indicated in Figure 5.4 (resulting in a role of prevalent “master frame” substantially played by frame $< t_a >$) is now totally lost in favour of a more balanced distribution of the tasks within the various involved all frames. Moreover, the entire set of frames now results to be “more bound” (by using a colloquial expression) than that of Figure 5.4, due to the redundant existence of additional feedback control actions.

At this point the same idea underlying the Automatic Independent Motion Control previously seen, can be easily extended to the case of Automatic

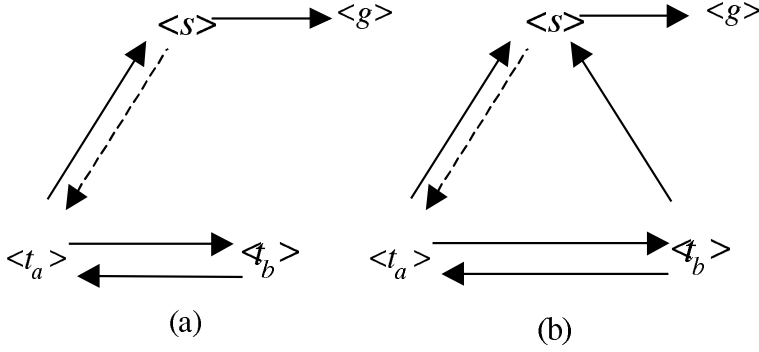


Figure 5.9. Frame tracking arrangement for the automatic mode

Coordinated Motion Control via the use of one of the conceptual schemes of Figure 5.9 (where the right figure is an improvement of the left).

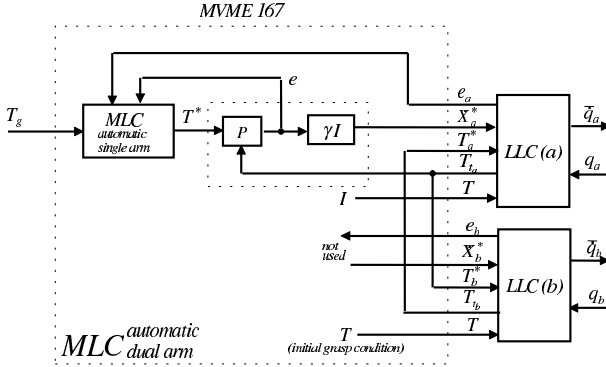


Figure 5.10. Control scheme for automatic coordinated motion, corresponding to the conceptual scheme of Figure 5.9(a)

To the above conceptual schemes there obviously correspond the functional ones of Figures 5.10, 5.11; which are in turn strictly analogous with the teleoperated ones of Figures 5.6, 5.8, but only differing for the internal presence of the functional block “MLC automatic single arm” in lieu of the teleoperated one.

By following the same identical philosophy which has been adopted in Section 5, note how the block “MLC automatic single arm” now receives as input both the “external” tracking error e and the “internal” one e_a .

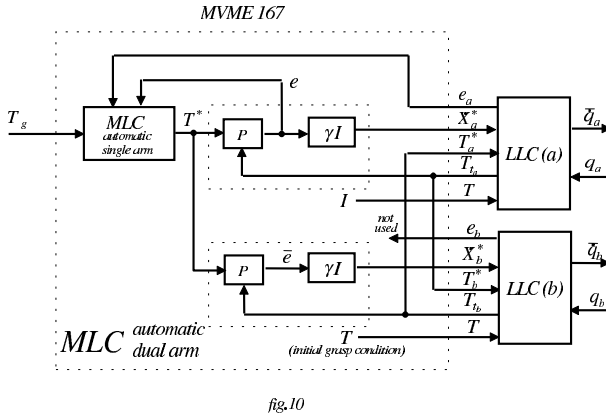


Figure 5.11. Control scheme for automatic coordinated motion, corresponding to the conceptual scheme of Figure 5.9(b)

5.3 Interaction Control during Coordinated Motion

In the previous section, the problem of moving a grasped object in a coordinated fashion has been considered without paying any attention to the aspect of possibly controlling, during the motion, also the interaction forces and torques which could actually exist in correspondence of the grasping zones of the transported object.

As a matter of fact, it could have been argued that the previously proposed control schemes substantially reflect the idea of actually “mimicking” the coordinated motion of a lightweight object; thus implying a transportation that, in presence of a real lightweight body, consequently occurs with (substantially) null interacting forces and torques.

In many applications this might actually be a requirement to be met (especially when delicate, other than lightweight, bodies need to be transported) even if, certainly, it does not correspond to the most general one. In fact, there could be also cases where, for some reasons, the transported object needs to be maintained stretched, or compressed, or even torqued, during motion. For such additional cases, we shall now show how the previously proposed “mimicking” control schemes can be used as sort of “skeletons”, upon which more proper control schemes can be built, also allowing interaction control, whenever needed.

To this end, let us restart by considering the initial grasping of a lightweight rigid object by part of both the manipulators, as depicted again in Figure 5.12.

In this new figure, force F and torque M represent the possible interaction vectors acting on the end effector of robot (a) and projected on the corresponding frame $\langle e_a \rangle$. In the same figure, the opposite of such force and

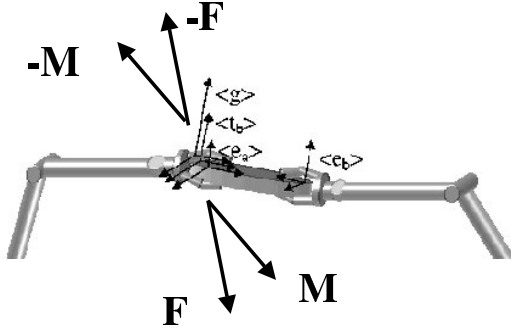


Figure 5.12. Forces and torques when the grippers are handling the object

torque vectors instead constitute the interactions acting on the end effector of robot (b) whenever translated on its own tool frame $\langle t_b \rangle$ coinciding (at least initially) with $\langle e_a \rangle$ as done in Section 5.2. Obviously enough, when existing, such interactions turn out to be one the opposite of the other, since they are referred to an assumed lightweight rigid body (then substantially with no mass and inertia). Moreover, since the rigid body is supposed to be firmly grasped, this implies that the geometric “internal errors” $e_a = -e_b$ necessarily remain at their initial null values even during coordinated motion, just as a consequence of the assumed firm grasping constraints.

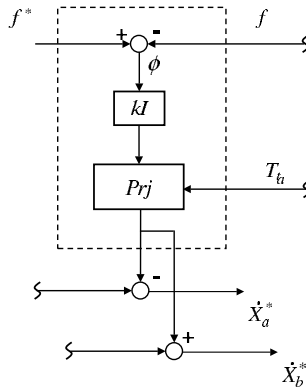


Figure 5.13. Control scheme to realise the corrected signals to add in the reference velocities control loop

Further, as pointed out at the beginning of this section, if the coordinated motion is performed as in Section 5.2, such motion will actually proceed with substantially null interactions. Now, provided that non-null (constant) refer-

ence vectors F^* and M^* are desired for the (otherwise almost null) interaction vectors F , M , the following very simple interaction feedback scheme can be used, in superposition with any one of the two of Section 5.2, as indicated.

In this scheme (Figure 5.13) the global interaction reference vector $f^* = [F^{*T} \ M^{*T}]^T$ is compared with the measured one $f = [F^T \ M^T]^T$ —indifferently provided by one of the robot force/torque sensors; for instance the one mounted on robot (a)— and the corresponding interaction error vector Φ distributed (after multiplication by a constant gain matrix kI and successive projection on the absolute frame $< 0 >$) to both the LLC blocks in terms of a couple of equal and opposite additional Cartesian velocity commands, non perturbing the overall coordinated motion. Notice that the proposed control loop is explicitly lacking of an integral action (generally appearing within schemes proposed in the literature) since such action is already provided “internally” by the presence of the existing PI joint velocity regulators located at the VLLC layer. Then, the proposed simple loop also guarantees the regulation to zero of the associated interaction error vector Φ . Additional details of the resulting overall coordinated control scheme can be found in [9]. Extensions of the overall coordinated motion and interaction control schemes to more general cases of slightly deformable, or even fully deformable, lightweight objects, are currently under development.

6 Comments on the Role of the High Level Control Layer

The previous section has brought into evidence how different types of independent and cooperative tasks can be accomplished via the use of associated control schemes, each one of them resulting from the suitable interconnection of a limited set of “primitive” or “basic” (in a sense) functional building blocks, which appear to be shared by all the task control scheme up to now considered.

It is the authors’ opinion that such sort of “composition property” could actually be maintained (possibly by slightly enlarging the set of basic functional blocks) even in the presence of any possible grow in the number and type of tasks which could be asked to be performed by the multiarm cell. Then, in this perspective we can consequently think of assigning to the HLC layer the function of activating the right MLC scheme upon request of execution of the corresponding task. Obviously enough, this can be done in many different ways, ranging from a straightforward one, just corresponding to the activation of only one among a finite number of separate programs or schemes (one per task), to the more involved, requiring the automatic extraction, within an enlarged superset of admissible possibilities, of the right interconnections among the basic functional blocks (i.e. the right organisation among a set of basic routines) upon request of execution of the corresponding task.

In case of few tasks to be activated (as it is for instance for the number of tasks, still very limited, considered in this work) the first solution actually reveals to be the simplest to be implemented and even the best performing one (just for this reason it corresponds to the one currently adopted within the referenced experimental system of Figure 1.1).

In the opposite case, the more complex involved tasks would reveal to be more advisable, since allowing much more flexibility in terms of modification, diversification, and/or addition of new tasks, without requiring any substantial coding effort, after the initial developing stage. Progress toward the realisation of an HLC layer of the second type should however proceed in parallel with the sought further progress in the construction of a structured framework for classes of cooperative tasks much wider than the one, still very simple, which has been considered in this work.

Another set of problems which must be taken into account in the construction of the HLC layer (in a form or in another) is represented by the possible “spike disturbances” which might be induced in the mechanical structure during the short duration of the switching time between different MLC schemes. As a matter of fact, in order to overcome such problems, particular attention must be paid to two complimentary aspects; namely: a), the theoretic conditions (in terms of input, output, and error values, plus logic conditions) which allow smooth transitions from different tasks and b), the need of not allowing any “empty” sampling interval during transition.

As a matter of fact, while the second aspect can only be handled by carefully acting at the Os level, the first one is a problem of analysing in details the possibility of directly sequencing (in the mentioned terms) different couples of tasks. In particular, the foreseen need of defining “auxiliary tasks” allowing the sequential connection of tasks, that otherwise cannot be sequenced, should also be kept into account as a possibility for further developments. Up to date, with reference to the experimental system of Figure 1.1 and for the class of tasks considered here, such analysis has been performed, almost trivially, on an inspection basis only.

From the above latter considerations a more precise characterisation of the role to be played by the HLC layer follows directly; namely that of an event based closed loop logic controller, managing the overall system. Just due to this, the cluster HLC+MLC naturally acquires characteristics which are proper of the so-called “Hybrid Control Systems”, thus allowing the correct positioning of the overall multiarm control and coordination problem within the framework of one of the most promising current research areas in the field of Automation.

7 Conclusions

Within this chapter, many of the aspects related with the development of a functional and algorithmic architecture for multiarm systems have been

addressed in some details. Special attention has been directed toward the definition of the roles to be played by the various layers in the perspective of reflecting, at each layer, a specific level of abstraction of the solution of the global command and control problem; thus also leading to an efficient yet flexible framework for the organisation of the interaction of relevant hardware/software components.

Particular emphasis has been given to the various basic functional blocks characterising each layer of the architecture, whose composition possibilities have revealed to be very useful in the perspective of being able to incrementally construct (via suitable composition rules) wider classes of complex cooperative tasks. Some comments concerning the relevant real-time Hw/Sw design aspects have also been given, together with those concerning further research directions to be undertaken.

References

1. 1998 Special Issue on Integrated Architectures for Robot Control and Programming. *International Journal of Robotics Research* 17(4)
2. Aicardi M, Caiti A, Cannata G, Casalino G 1995 Stability and robustness analysis of a two layered hierarchical architecture for the control of robots in the operational space. In: *Proceedings of 1995 IEEE International Conference on Robotics and Automation* Nagoya, J, pp 2771–2778
3. Aicardi M, Cannata G, Casalino G 1995 Task space robot control: Convergence analysis and gravity compensation via integral feedback. In: *Proceedings of 35th IEEE Conference on Decision and Control* Kobe, J, pp 3032–3037
4. Aicardi M, Cannata G, Casalino G 1996 Task level position control of robots, and static errors compensation. In: *Proceedings of 2nd World Automation Congress* Montpellier, F, pp 17–27
5. Alami R, Chatila R, Fleury S, Ghallab M, Ingrand F 1998 An architecture for autonomy. *International Journal of Robotics Research* 17:315–337
6. Borelly J, Coste-Maniere E, Espiau B 1998 The Orcad architecture. *International Journal of Robotics Research* 17:338–359
7. Casalino G 1996 Hierarchical dynamic control of robotic manipulators. In: *Notes from the course lessons in Industrial Robotics* (in Italian) Faculty of Engineering, University of Genova
8. Casalino G 1996 Task composition within planning and control of robotic manipulators. In: *Notes from the course lessons in Industrial Robotics* (in Italian) Faculty of Engineering, University of Genova
9. Casalino G 2000 *Remarks on Force/Torque Control Performed via Cartesian Velocity Commands* DIST Technical Memo, University of Genoa
10. Casalino G, Angeletti D, Bozzo T, Marani G 2001 Dexterous underwater object manipulation via multirobot cooperating systems. In: *Proceedings of 2001 IEEE International Conference on Robotics and Automation* Seoul, KOR
11. Casalino G, Angeletti D, Cannata G, Marani G 1999 Symbolic based model builder for multibody systems. In: *5th International Conference on Information Systems, Analysis and Synthesis* Orlando, FL

12. Casalino G, Angeletti D, Cannata G, Marani G 2000 On the functional and algorithmic control of the Amadeus dual arm robotic workcell. In: *6th World Automation Congress* Maui, HI
13. Casalino G, Angeletti D, Cannata G, Marani G 2000 Design of functional and algorithmic control architectures for multirobot systems. *Preprints of 6th IFAC Symposium on Robot Control* Vienna, A, pp 109–114
14. Casalino G, Angeletti D, Cannata G, Marani G 2001 The functional and algorithmic control architectures for the Amadeus multirobot workcell. *IEEE Journal of Oceanic Engineering* 26
15. Coste-Maniere E, Espiau B 1998 Editorial to Special Issue on Integrated Architectures for Robot Control and Programming. *International Journal of Robotics Research* 17(4):i–ii
16. Craig J J 1989 *Introduction to Robotics: Mechanics and Control* (2nd ed) Addison Wesley, Reading, MA
17. Marani G, Casalino G 1999 Robotics developer studio. In: *ANIPLA National Congress on Automation* Roma, I
18. Nakamura Y 1991 *Advanced Robotics: Redundancy and Optimization* Addison Wesley, Reading, MA
19. Schneider S A, Chen V W, Pardo-Castellote G, Wang H H 1998 Controlshell: A software architecture for complex electromechanical systems. *International Journal of Robotics Research* 17:360–380

Modelling and Control of Servomechanisms

Gianni Ferretti, GianAntonio Magnani, and Paolo Rocco

Dipartimento di Elettronica e Informazione
Politecnico di Milano

<http://www.elet.polimi.it/section/automeng/control>

Two major sources of load oscillations in servo systems are dealt with in this contribution, namely pulsating torque disturbances and torsional flexibility. After reviewing the state of the art in torque ripple compensation, a compact model of torque disturbances is presented and algorithms are given for the identification of model parameters. Thanks to its compactness, the model allows the implementation of a straightforward and effective disturbance compensation technique. Torsional flexibility is modelled by a classical two-mass model with spring and damper. A short review on the control of elastic joints is given. Then a careful analysis of the properties of the two-mass model is performed, leading to the design of a P/PI controller where suppression of load oscillations rather than fast setpoint tracking of the motor position is pursued. Further improvements of the load behaviour can be achieved by a notch filter placed outside the velocity loop. Finally, the most complete solution available with only motor position measurements, an LQG controller with feedforward actions for load setpoint tracking, is investigated. Experimental and simulation results are given to assess the effectiveness of the proposed approach.

1 Introduction

Positioning servomechanisms are used in a large number of applications in robotics, machine tools, packaging, printing and textile machines, entertainment products, computer peripherals and devices, space and defense pointing, motion systems and other applications. The large majority of servos use permanent magnets motors connected to the load by a transmission chain (or gearbox), and a single position sensor, either an encoder or a resolver, mounted on the motor shaft. This is by far the most common solution adopted in articulated robotic manipulators.

By feeding back the motor position only, it is relatively easy to obtain satisfactory control of the motor position and velocity, using either P/PI or PID control, or more advanced compensation techniques [15], [24], [30], [29], [32]. However, this does not guarantee a satisfactory control of the load position and velocity, for demanding motion control applications, in particular during slow motion. For instance, oscillations of the tip of a manipulator arm or of the tool of a milling machine may arise. A large variety of sources of oscillation can be identified: motor [5], [13], [20] and gearbox [10] pulsating torque

disturbances, torsional elasticity of the transmission chain, sensor noise [11], friction [2], backlash, and others. Among them, special attention should be paid to the transmission elasticity, which generally causes the lowest (“first”) resonance frequency of the positioning system, and to the motor torque disturbances (ripple), that mostly excite the resonance at low velocities [7].

This paper deals first with the compensation of motor torque disturbances, and then with the design of the “standard” P/PI controller, carried out emphasising a well damped behaviour of the load rather than a fast setpoint tracking of the motor position.

As far as the torque ripple suppression is concerned, several approaches have been proposed [4], [12], [13], [16]: some of them will be reviewed in Section 2.2. In the present contribution, the problem is addressed starting from the formulation of a compact model of the torque pulsations in sinusoidal permanent magnet AC brushless motors (PMAC BLM) [5]. The model defines the torque delivered by the motor as a scalar function of the rotor position and avoids addressing the behaviour of each phase for the identification of its parameters.

Methods and algorithms are then proposed for the identification of the disturbance model, starting from spectral identification methods. They require a controlled motion at constant low velocity: information about the torque pulsations is extracted from the output of the closed loop controller and correlated to the angular position of the rotor. Then, in order to cope with the time varying nature of the fundamental harmonic of the disturbance, a methodology for online adaptation of the parameter estimates is proposed.

A compensation technique is discussed as well. The idea is to modify online the current reference produced by the position controller as a function of the rotor position. As such, the algorithm can be performed directly in the position controllers, even though modifications of the lookup tables used to compute the brushless functions for the single motors would be feasible as well. Experimental results are given to show the effectiveness of the compensation technique.

Some existing approaches to the control of elastic joints will be briefly reviewed in Section 3.2. As a matter of fact, load vibrations can be reduced by a proper design of the P/PI controller, based on a careful analysis of the properties of the two-mass model of the elastic system. As a consequence of the relative position of poles and zeros of the process transfer function, it is straightforward to show, using the root locus analysis, that the common practice of increasing the bandwidth of the velocity loop (high gain of the PI regulator) fosters the oscillation of the load. It is then suggested a way to choose the loop gain, trading off between higher values, for a fast velocity response, and lower values, for more damped oscillations of the load. Further improvement of the damping of the closed loop dominant poles can also be obtained with a proper tuning of a notch filter that, however, has to be placed outside the velocity loop.

Modern control theory suggests that the most complete solutions for the control of the system consists in feeding back the estimates of the whole state of the system, obtained through suitable observers or Kalman filters [1]. This solution has been adopted in the past [21] and is investigated here, emphasising the choice of the closed loop poles, based on control effort and sensitivity considerations, and the design of the feedforward blocks for accurate tracking of the load position setpoint.

Modelling, identification and compensation of pulsating torque disturbances are dealt with in Section 2, where experimental results are also given. Section 3 discusses first the properties of the two-mass model, then deals with the P/PI design and finally with the notch filter design. Experimental and simulation results are given. Section 4 presents the design and experimental validation of the LQG plus feedforward controller.

2 Modelling and Compensation of Torque Disturbances

2.1 Torque Disturbance Generation

Consider the functional scheme of a sinusoidal PMAC machine, represented in Figure 2.1.

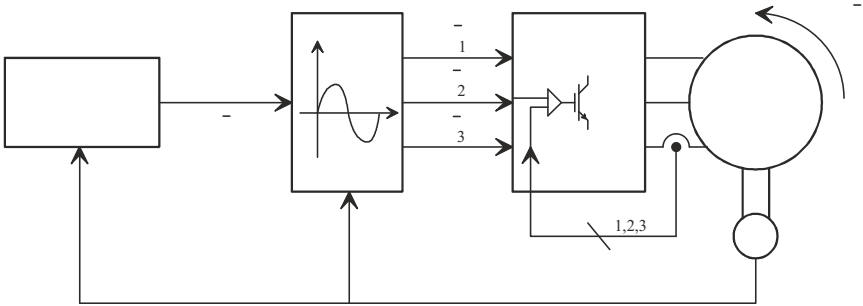


Figure 2.1. Functional scheme of a brushless motor

If a reference torque $\bar{\tau}$ should be delivered by the motor, typically as required by a position controller, the current reference \bar{I} has to be given the value $\bar{I} = \bar{\tau}/K_t$, where K_t is the torque constant. This scalar setpoint is then modulated through three sinusoidal functions of the electrical angle $\alpha = pq_m$, p being the number of pole pairs and q_m being the motor angle, that are offset by $2\pi/3$ one from each other. The three resulting signals become the current setpoints for the three phases. High bandwidth current controllers make the

currents track their setpoints in each phase. If the current reference in each phase is given the same dependence on the electrical angle characterising the back EMF (ideally sinusoidal or trapezoidal), a torque τ is produced, approximately equal (in a band of frequencies limited by the current loops) to the desired torque $\bar{\tau}$, and thus proportional to the scalar current reference \bar{I} .

The behaviour of a real motor differs however from the simple one here described. Several constructive imperfections of the motor and the servodrive sum up to form a pulsating disturbance on the torque. Cogging torque and pulsating torque due to offsets in the current drives are examples of pulsating torque disturbances not related to the current reference. Cogging is mainly due to the presence of slots in the stator, where the rotor generated magnetic flux seeks a minimum reluctance path. It is present even when the drive is not connected to the power supply and depends in a periodic way on the mechanical angle. When the drive is connected to the power supply and the current reference is zero, a current offset may be present in one or both the closed loop controlled phases, and thus on the third one. These currents generate an offset torque periodic with period 2π with respect to the electrical angle α .

Pulsating torques related to the current reference are due to imperfections in the construction of the motor and the drive, implying that both the back EMF profiles and the phase currents may be affected by undesired higher order harmonics, while obviously maintaining the periodicity of 2π with respect to the electrical angle.

2.2 Existing Solutions for Torque Minimisation

A complete review of the methods used to minimise torque disturbances is reported in [20]. The techniques are classified into two main categories: motor-based, where the fundamental electromagnetic sources of the disturbance are addressed, and the design of the motor is adjusted in order to minimise them, and control-based, where active control schemes modify the excitation, to correct for any of the nonideal characteristics of the machine or its associated power inverter. Among this second class of techniques (the most pertinent with the present contribution) is the work by Jahns [19], which proposes compensation of the disturbance effects through high bandwidth speed feedback. Selective elimination of torque ripple harmonics, through proper injection of harmonics of suitable orders in the current profiles, was first proposed in [16] and further extended in [4], where an iterative procedure is worked out to eliminate higher order harmonics in the torque. In [12] and [14] the torque disturbance minimisation is treated as a constrained optimisation problem: the current profiles that minimise suitable functionals (related to the ohmic losses in the windings) and guarantee elimination of the torque harmonics up to a predetermined order, together with satisfaction of other constraints (such as Y connection of the phases), are computed, based on the known harmonics

of the back EMF profiles. Inversion of the back EMF profiles of the single phases (computed by means of finite element analysis) is proposed in [3]. Recently, Holtz and Springob [13] have proposed a self-commissioning control scheme which identifies the machine parameters and adaptively controls the current using a standard microcontroller.

2.3 Torque Disturbance Model

As it is shown in [5], the following relation can be used to represent in a compact form the effects of the disturbances on the torque production:

$$\tau = \tau(\alpha, \bar{I}) = \gamma(\alpha) + K_t \bar{I}(1 + \delta(\alpha)). \quad (2.1)$$

The term $\gamma(\alpha)$ accounts for the disturbances due to the cogging torque and to the current offset in the drives, while the second term is responsible for the nominal torque (with $\delta(\alpha) = 0$) and for the disturbances related to the harmonic content. It is also possible to include in $\delta(\alpha)$ the effects of the amplitude imbalance and the phase misalignments of the current and back EMF shapes profiles [5].

The functions $\gamma(\alpha)$ and $\delta(\alpha)$ in (2.1) are time-invariant, periodic with period 2π with respect to the electrical angle and null in nominal conditions. Notice that the formulation of the model does not require knowledge of the characteristics of the single phases (in terms of back EMF profiles, imbalance and misalignments), but only of their combined effect in the parameters of the scalar equation (2.1).

Fourier expansions of the functions $\gamma(\alpha)$ and $\delta(\alpha)$ are introduced in order to parametrise the model, in the following form:

$$\gamma(\alpha) = \sum_{k \in N_\gamma} \gamma_k \sin(k\alpha + \beta_k) \quad (2.2)$$

$$\delta(\alpha) = \sum_{k \in N_\delta} \delta_k \sin(k\alpha + \psi_k) \quad (2.3)$$

where N_γ and N_δ are the sets of harmonics to be considered in the expansions of $\gamma(\alpha)$ and $\delta(\alpha)$, respectively. A validation of the model, performed through several static measurements of the torque of a motor mechanically constrained in different positions and subject to different current references, has been described in detail in [5].

2.4 Spectral Identification of the Disturbance

For the purpose of online compensation, the harmonics of the torque must be obtained from the motor through simple motion experiments. With reference to expressions (2.2,2.3), the goal of the identification is therefore to give estimates for the amplitudes γ_k and δ_k and phases β_k and ψ_k . The basic idea

is to get them by examining the output of the position controller in a closed loop experiment, where a constant velocity motion, at a very slow velocity, is commanded.

Consider first the block diagram (Figure 2.2) of the motor controlled by a linear regulator, where the disturbance is represented as an external input (actually, according to (2.1), it depends on α and \bar{I}), while $G(s)$ and $R(s)$ stand for the transfer functions of the process and of the regulator, respectively.

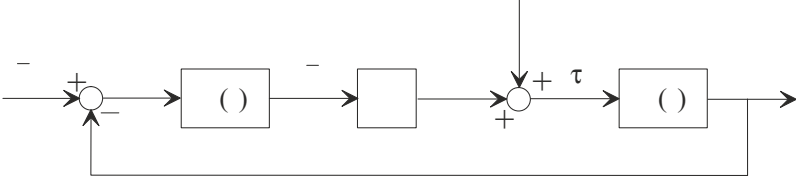


Figure 2.2. Position controlled motor with torque disturbance

2.4.1 Identification of $\gamma(\alpha)$. A practical way to cope with the identification problem is to realise that the amplitude of the disturbance term proportional to the current reference $K_t \bar{I} \delta(\alpha)$ becomes actually comparable to the amplitude of the other disturbance term $\gamma(\alpha)$ only when the current reference \bar{I} takes high values, say more than 30% of the value corresponding to the nominal torque. Since in a constant and low velocity motion the main reason for the torque to take high values is to counteract a constant load (such as the gravity load or a particularly high friction term), experimental configurations have to be found, where these terms are minimised. In these configurations, the torque disturbance can be entirely ascribed to the current-independent term $\gamma(\alpha)$, with quite good approximation.

The following relation, in terms of transfer functions, is easily obtained:

$$K_t \bar{I} = \frac{K_t R(s)}{1 + L(s)} \bar{q}_m - \frac{L(s)}{1 + L(s)} d \quad (2.4)$$

where $L(s) = K_t R(s) G(s)$, and

$$G(s) = \frac{1}{J_m s^2 + D_m s}$$

with J_m and D_m being the inertia and viscous friction coefficient of the motor respectively. Inverting (2.4) one obtains:

$$d = \frac{1}{G(s)} \bar{q}_m - \frac{1 + L(s)}{L(s)} K_t \bar{I} \quad (2.5)$$

and taking the following approximation for the closed loop transfer function of the system:

$$\frac{L(s)}{1 + L(s)} \cong \frac{1}{1 + s/\omega_c}$$

with ω_c being the crossover frequency of the loop, (2.5) can be rewritten as follows:

$$d \cong (J_ms^2 + D_ms)\bar{q}_m - (1 + s/\omega_c)K_t\bar{I}$$

finally yielding, in the time domain:

$$d \cong J_m\ddot{\bar{q}}_m + D_m\dot{\bar{q}}_m - K_t\bar{I} - \frac{K_t}{\omega_c}\dot{\bar{I}}. \quad (2.6)$$

The identification thus goes like this: assuming the position reference a known function of time, its first two derivatives are computed at each sampling instant, and the samples of the current reference commanded by the position regulator are stored. The time derivative of this signal is then computed numerically, while (2.6) gives the time series of the estimates of the disturbance. Since $d \cong \gamma(\alpha)$, Fourier analysis of this time series with respect to the stored position measurements of the angle q_m provides estimates of the amplitudes and phases of the harmonics of $\gamma(\alpha)$.

Notice that the time derivative of the current reference required by (2.6) can yield noise amplification. If this is the case, this latter term can be neglected, provided that the spectrum of the position reference signal is entirely contained in the bandwidth of the closed loop system.

2.4.2 Identification of $\delta(\alpha)$. For the identification of the function $\delta(\alpha)$, a second experiment has to be performed, in a configuration where a significant static load is present, so that the contributions of the terms $\gamma(\alpha)$ and $\delta(\alpha)$ to the disturbance are actually comparable. The disturbance d is still computed as in (2.6), but now it is assumed:

$$d = \gamma(\alpha) + K_t\bar{I}\delta(\alpha).$$

Fourier analysis of the function d derived from the above relation, while using the approximation of $\gamma(\alpha)$ already known from the previous identification, will provide estimates of δ_k and ψ_k .

2.5 Adaptive Identification of the Disturbance

While the procedure described in Section 2.4 solves the identification problem with a modest computational effort, it is still a batch method, where the set of data required to compute (2.6) is first obtained, perhaps before normal operating condition of the motor, and then the result of the spectral identification are applied for compensation of the disturbance.

However the first harmonic of the current independent component $\gamma(\delta)$ shows typically a time dependence [6], due to a thermal drift of the offset in

the current sensors. In order to cope with this and other time dependences, an adaptation of the estimates is to be implemented. Here the procedure for an adaptive estimation of the first harmonic of the disturbance $\gamma(\alpha)$ will be shortly outlined.

The idea is simply to find the estimates for amplitude γ_1 and phase β_1 of the sinusoid by minimising the following quadratic loss function:

$$J = \sum_{i=1}^N \left[d(i) - \hat{\gamma}_1 \sin(\alpha(i) + \hat{\beta}_1) \right]^2$$

where N is the total number of data used for the identification and d is evaluated as in (2.6). Explicit formulas for the estimates of γ_1 and β_1 can be easily obtained with this minimisation. A recursive algorithm can then be setup in the form:

$$\hat{\beta}_1(i) = \hat{\beta}_1(i-1) + f_\beta(\hat{\beta}_1(i-1), d(i)) \quad (2.7)$$

$$\hat{\gamma}_1(i) = \hat{\gamma}_1(i-1) + f_\gamma(\hat{\gamma}_1(i-1), \hat{\beta}_1(i), d(i)). \quad (2.8)$$

Details on functions f_β and f_γ can be found in [9].

2.6 Design of a Compensation Technique

Once the estimates for functions $\gamma(\alpha)$ and $\delta(\alpha)$ have been obtained, either from the spectral analysis outlined in Section 2.4 or the recursive least squares algorithm sketched in Section 2.5, a compensation scheme as in Figure 2.3 can be setup. Let \bar{I}^o be the current reference determined on the basis of a model

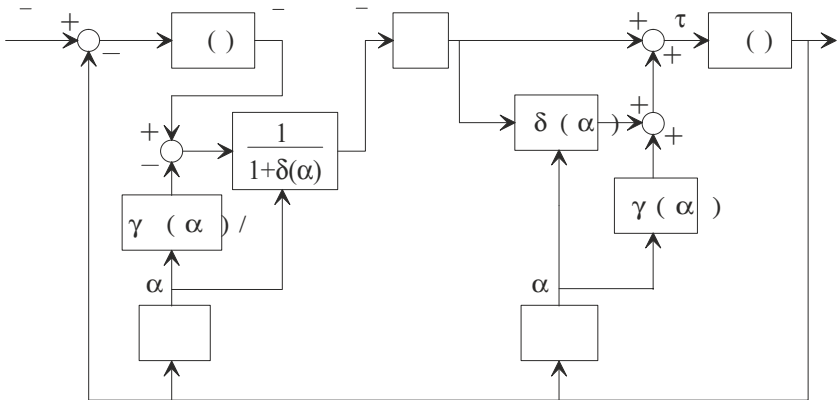


Figure 2.3. Position controlled motor with compensation

of the motor free of the disturbances. In other words, let $\bar{I}^o = \bar{\tau}/K_t$, where $\bar{\tau}$ is the torque that the motor is supposed to deliver. The actual current reference will be determined as follows:

$$\bar{I} = \frac{\bar{I}^o - \hat{\gamma}(\alpha)/K_t}{1 + \hat{\delta}(\alpha)}.$$

This way, in nominal conditions ($\delta(\alpha) = \hat{\delta}(\alpha)$, $\gamma(\alpha) = \hat{\gamma}(\alpha)$) the effect of the disturbance is completely eliminated.

Thus the estimates $\hat{\gamma}(\alpha)$ and $\hat{\delta}(\alpha)$ are used to continuously (i.e. at every sampling instant) modify the current reference \bar{I}^o produced by a closed loop controller.

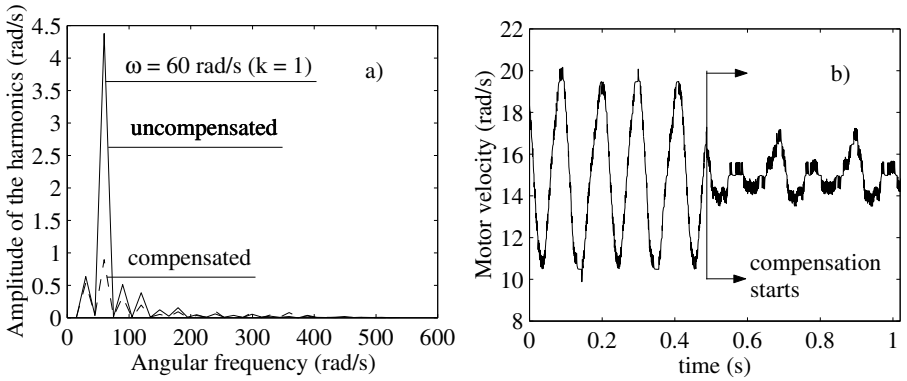


Figure 2.4. a) Amplitudes of the harmonics of the motor velocity; b) Time history of the velocity

2.7 Experimental Results

Experiments have been made on an ECS (Electronic Control Systems) 105ES 30022 PMAC BLM (nominal torque 2.2 Nm, 4 pole pairs), instrumented with a resolver, and coupled to a ISODRIVE 325 drive (by ECS), performing current control. The motor position control is performed through a PC, implementing a P/PI algorithm, with a sampling time of 512 μ s, as well as the identification algorithm. Embedding the software in the control hardware of the drive would require a memory storage which is estimated in about 2 Kbytes, and the computation of the trigonometric functions needed for the spectral analysis.

In the experiments the motor was free from external loads (including gravity): this means that only the current-independent term $\gamma(\alpha)$ was of

interest. Moreover, from the results of the spectral identification it became evident that only the harmonics corresponding to $k = 1, 6$ could have been retained from the complete expansion (i.e. $N_\gamma = \{1, 6\}$), without considerable loss of accuracy.

To assess the effectiveness of the proposed compensation scheme, an experiment has been run twice, without compensation and with the compensation on respectively, commanding a 15 rad/s constant velocity motion. The motor velocity, derived from the position measurements, has been Fourier analysed with respect to time before and after compensation, for validation purposes. The amplitudes of the harmonics are compared in Figure 2.4.a, where it is apparent that the harmonic at 60 rad/s (corresponding to $k = 1$) has been reduced to about 15% of its value without compensation. The other harmonics are left almost unchanged. Figure 2.4.b reports two consecutive records of the time histories of the velocities, the first one without compensation, the second one with compensation. Again it is apparent that the most evident first harmonic (whose period amounts to $2\pi/60 \cong 0.1$ s) is almost suppressed.

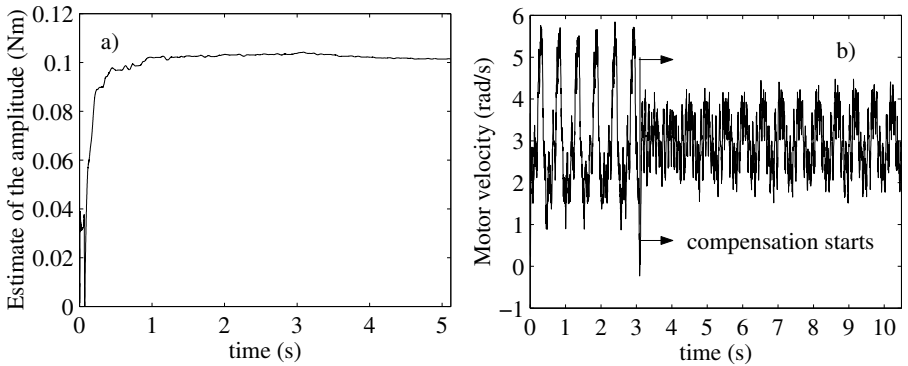


Figure 2.5. a) Evolution of the estimate for a constant velocity motion; b) Time history of the velocity

The recursive algorithm (2.7), (2.8) has then been used to implement an adaptive compensation of the first harmonic of the $\gamma(\alpha)$ disturbance. In the first part of the experiment, the compensation is kept off and the recursive algorithm works to update the estimates of the amplitude and phase of the disturbance. When these estimates reach a steady value, the compensation is switched on, while the recursive algorithm keeps updating the estimates.

First an experiment with a constant 3 rad/s velocity motion has been run. The evolution of the estimate for the amplitude of the harmonic is shown in Figure 2.5.a (the estimate of the phase follows a similar evolution). After

about 3 seconds the estimate reaches a steady value and the compensation is switched on. Figure 2.5.b reports the motor velocity, as numerically computed from the position measurements. The role of the compensation in suppressing the disturbance harmonic is quite evident.

A second experiment with a sinusoidal velocity reference has been run. Again, the estimate of the amplitude of the disturbance (Figure 2.6.a) converges after about 3 seconds (the same for the estimate of the phase). The motor velocity plotted in Figure 2.6.b emphasises again the reduction of the disturbance.

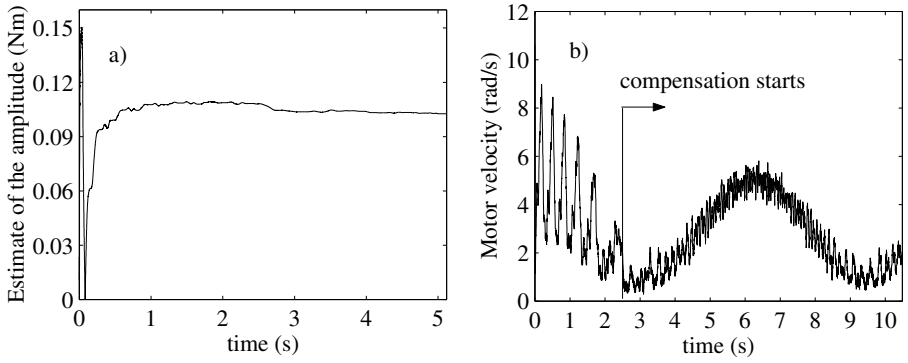


Figure 2.6. a) Evolution of the estimate for a sinusoidal velocity motion; b) Time history of the velocity

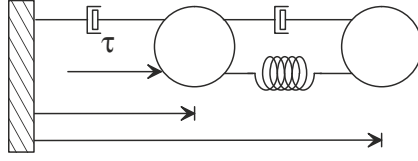
3 Modelling and Compensation of Load Oscillations

3.1 Model of an Elastic Servo Mechanism

A fast dynamics servo may have several flexible elements and connections, resulting in a dynamic behaviour affected by several resonances. From the point of view of the control design, it is essential to model correctly the lowest frequency resonance, which play an essential role in the design, as it is shown later on. Under certain conditions that are frequently satisfied, the first resonance frequency is correctly predicted by the well known model of two masses connected through an elastic element (Figure 3.1). The model is described by the following linear equations [27], [31], if the non linear friction terms are neglected:

$$\tau_m = K_t I \quad (3.1)$$

$$\tau_m = J_m \ddot{q}_m + D_m \dot{q}_m + \tau_t \quad (3.2)$$

**Figure 3.1.** Elastic servo positioning system

$$\tau_t = K_{el} (q_m - nq_l) + D_{el} (\dot{q}_m - n\dot{q}_l) \quad (3.3)$$

$$n\tau_t = J_l \ddot{q}_l \quad (3.4)$$

where τ_m is the motor torque, I is the motor current (actually the amplitude of the sinusoidal current of each stator phase of the brushless motor); K_t is the motor torque constant, q_l and q_m are the load and motor angular positions, respectively; J_l , J_m are the load and rotor inertia respectively; D_m is the viscous friction coefficient at the motor side; K_{el} , D_{el} , are the joint stiffness and damping factors, respectively; n is the transmission ratio; $n\tau_t$ is the torque delivered by the gear reducer at the load side. Furthermore, thanks to the fast dynamics current amplifiers, the difference between the current command \bar{I} (control variable) and the actual current I can be neglected ($\bar{I} = I$) for the purpose of position control design. Pulsating torque disturbances act as additive, variable frequency disturbances to be rejected by the control system. The transfer function $G_{vm}(s)$ from the motor torque to the motor velocity is then given by:

$$G_{vm}(s) = \frac{J_{lr}s^2 + D_{el}s + K_{el}}{\Delta(s)} \quad (3.5)$$

with

$$\begin{aligned} \Delta(s) = & J_{lr}J_ms^3 + [(J_m + J_{lr})D_{el} + J_{lr}D_m]s^2 + \\ & + [(J_m + J_{lr})K_{el} + D_mD_{el}]s + D_mK_{el} \end{aligned}$$

where $J_{lr} = J_l/n^2$ is the load inertia reported to the motor side.

The polynomial $\Delta(s)$ does not have a simple factorisation but, for common values of the system parameters, it shows a couple of complex roots whose natural frequency and damping factor can be approximately obtained letting $D_m = 0$:

$$\omega_p \cong \sqrt{\frac{J_{eff}K_{el}}{J_{lr}J_m}} \quad \xi_p \cong \frac{1}{2}D_{el}\sqrt{\frac{J_{eff}}{J_{lr}J_mK_{el}}}$$

where $J_{eff} = J_m + J_{lr}$. The third pole s_r is in a lower frequency range, as it is strongly related to the rigid behaviour of the system ($s_r \cong -D_m/J_{eff}$).

The zeros of the transfer function are complex too, with natural frequency and damping factor given by:

$$\omega_z = \sqrt{\frac{K_{el}}{J_{lr}}} \quad \xi_z = \frac{1}{2} D_{el} \sqrt{\frac{1}{J_{lr} K_{el}}}.$$

It is worth noting that

$$\frac{\omega_p}{\omega_z} = \frac{\xi_p}{\xi_z} = \sqrt{1 + \rho} > 1$$

$\rho = J_{lr}/J_m$ being the inertia ratio. Motor inertia and transmission ratio are frequently selected to pursue the inertia matching, namely to obtain $\rho = 1$.

3.2 Existing Solutions for the Compensation of Load Oscillations

In the review work [25] the disturbance observer technique, already presented in [30] and [29], is used to control a rigid servo affected by an unknown load torque, while the resonance ratio control is used when an elastic transmission is considered. The disturbance observer is also used in [15], [32], where torsional flexibility of the joint is not directly accounted for. A review of several works dealing with suppression of torsional oscillations is reported in [31], where the contributions are divided in three categories: 1) methods that exploit measurements on both the motor and the load sides; 2) methods based on the motor measurements only and observation of the state of the system; 3) notch filters plus conventional loopshaping techniques. The first category include most of the works where nonlinear control theory is used: singular perturbation and integral manifold theory is used in [22] for a single joint case and in [27] for the extension to a complete articulated manipulator; feedback linearisation is used in [8]; nonlinear observer theory in [28]; adaptive control theory in [18]. The second group includes more industry-oriented works, such as [17] and [26], as well as the early work [23], where a PD controller on the motor coordinate, enforced with a compensation of the gravitational effect, is considered. Finally, the analysis in [31] mostly belongs to the third group.

3.3 Classical Control and Load Oscillations

The control system requirements concern setpoint tracking and rejection of load and torque disturbances. Both of them demand for large control system bandwidth, namely for large feedback loop gain. However, a critical problem, especially during slow motion and at motion starts and stops, is due to load oscillations. For instance, load oscillations may reveal as vibrations of the end effector of a robot arm during the execution of a slow motion, like in arc welding tasks, while in milling and grinding machines they may cause some slight undulation, and thus poor finishing, of working surfaces. Since the analysis of the properties of the elastic model shows that high feedback gains on motor position increase load oscillations, a trade off is needed on the loop gain.

In most commercial products, position control (Figure 3.2) consists of a position loop with a proportional (P) regulator, implemented for instance in a CNC, cascaded with a velocity loop with proportional-integral (PI) regulator, implemented in the drive electronics. The integral action ensures that the error on motor position for constant setpoint, load torque and torque disturbances (τ_d), vanishes at steady-state. While in the past the velocity was sensed by a tachometer, in current products it is obtained by numerical differentiation of the motor position, sensed by either an encoder or a resolver. To improve the setpoint tracking capabilities, a feedforward derivative action is also inserted from the position setpoint to the velocity one.

It is easy to check that this control scheme is equivalent to a PID controller with real zeros, fed with the motor position error $e_p = \bar{q}_m - q_m$ and with output \bar{I} .

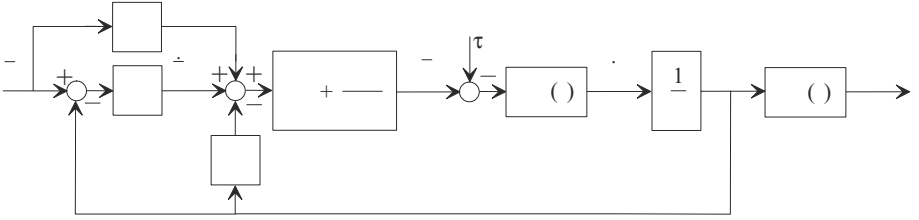


Figure 3.2. P/PI control with feedforward action

Note that in the block diagram of Figure 3.2 a block has been added, with transfer function:

$$G_{lm}(s) = \frac{1}{n} \frac{D_{el}s + K_{el}}{J_{lr}s^2 + D_{el}s + K_{el}}, \quad (3.6)$$

which allows computation of q_l from q_m .

The velocity loop is designed first. The loop transfer function is:

$$L_v(s) = K_{cv}K_tJ_{lr} \frac{(s + 1/T_{iv})(s^2 + 2\xi_z\omega_zs + \omega_z^2)}{s\Delta(s)}$$

and the closed loop transfer function from setpoint to motor velocity is:

$$F_v(s) = K_{cv}K_tJ_{lr} \frac{(s + 1/T_{iv})(s^2 + 2\xi_z\omega_zs + \omega_z^2)}{\Delta_v(s)}$$

where $\Delta_v(s)$ is defined as

$$\begin{aligned} \Delta_v(s) = & J_{lr}J_ms^4 + [(J_m + J_{lr})D_{el} + J_{lr}D_m + K_{cv}K_tJ_{lr}]s^3 + \\ & + [(J_m + J_{lr})K_{el} + D_mD_{el} + K_{cv}K_tJ_{lr}(2\xi_z\omega_z + 1/T_{iv})]s^2 + \\ & + [D_mK_{el} + K_{cv}K_tJ_{lr}(\omega_z^2 + 2\xi_z\omega_z/T_{iv})]s + K_{cv}K_tJ_{lr}\omega_z^2/T_{iv}. \end{aligned}$$

For $K_{cv} \rightarrow \infty$ the system will be stable, whatever $T_{iv} > 0$, but two poles approach the lightly damped process complex zeros. In this case, also two roots of the characteristic polynomial of the closed position loop, given by:

$$\Delta_p(s) = s\Delta_v(s) + K_{cp}K_{cv}K_tJ_{lr}(s + 1/T_{iv})(s^2 + 2\xi_z\omega_zs + \omega_z^2)$$

become the process zeros, whatever $K_{cp} > 0$. On the other hand, for smaller values of K_{cv} two of the position loop poles become the process zeros only for $K_{cp} \rightarrow \infty$.

If two closed loop poles are equal (or almost equal) to the process zeros, the relevant dynamics is not observable from the motor position, but it reveals with oscillations of the load, because of the poles of $G_{lm}(s)$ (3.6).

As a conclusion, the controller design requires to trade off between higher gains (in particular K_{cv}), that increase setpoint tracking and torque disturbance rejection capabilities, and lower gains, needed to keep the damping of closed loop poles reasonably high.

In this respect, the common industrial practice of tuning the velocity loop as fast as possible, increasing the gain K_{cv} until an “audible noise” is generated by the motor, proves to be quite dangerous. In fact, this is exactly the way to induce load oscillations, as the high gain velocity feedback places two closed loop poles near the lightly damped process zeros, where they remain also after closing the position loop.

3.3.1 An Illustrative Example. The prototype servo positioning system of Figure 3.3 is considered as an illustrative example.

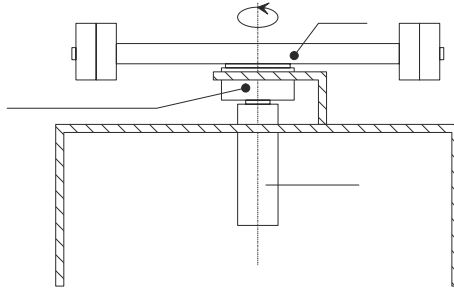


Figure 3.3. Prototype positioning system

It consists of a Control Techniques permanent magnet AC brushless motor (PMAC BLM), with a nominal torque of 2.3 Nm and a nominal power of 700 W, a Harmonic Drive speed reducer, and a load. Since the load rotates in a horizontal plane the gravity force does not act on the system. This system is controlled by a PC as described in Section 2.7. The parameters of the testbed are given in Table 3.1 (in SI units).

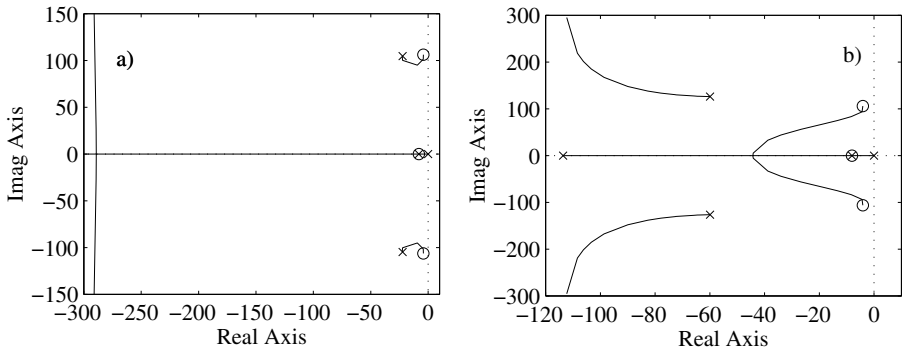


Figure 3.4. a) Root locus for $K_{cv} = 5.2 \times 10^{-2}$; b) Root locus for $K_{cv} = 1.84 \times 10^{-2}$

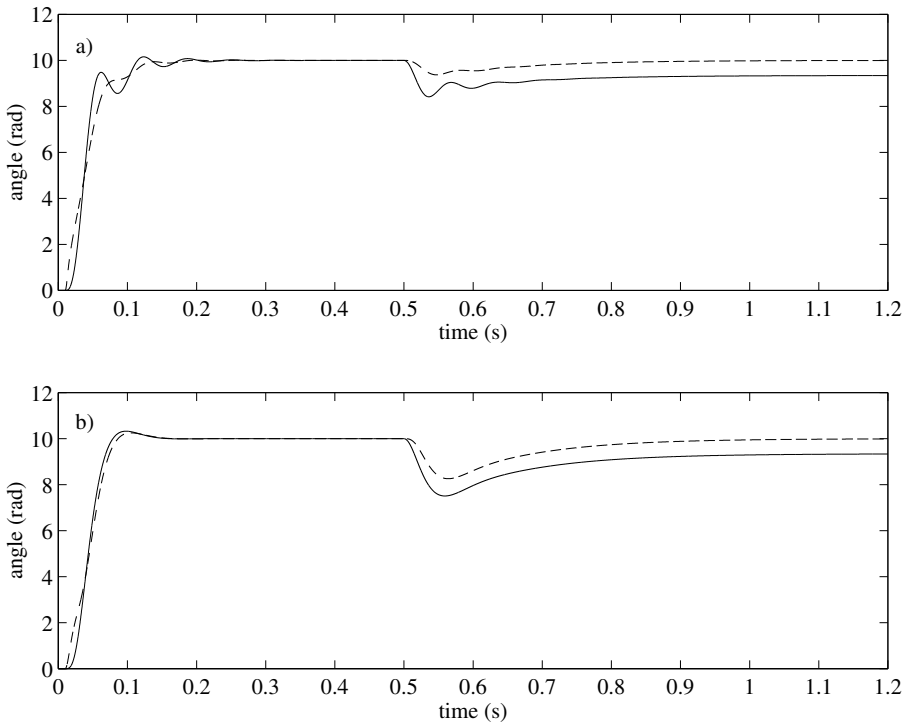


Figure 3.5. Time responses of q_m (dashed line) and nq_l (solid line) for a) $K_{cv} = 5.2 \times 10^{-2}$ and b) $K_{cv} = 1.84 \times 10^{-2}$

K_t	J_m	J_l	n	D_m	K_{el}	D_{el}
1.6	1.5×10^{-4}	2.7	100	3.4×10^{-3}	3.05	2.2×10^{-3}
ω_z	ξ_z	ω_p	ξ_p	$ s_r $	T_z	
106.3	3.8×10^{-2}	177.5	$0.105 \div 0.064$	8.13	7.2×10^{-4}	

Table 3.1. Physical parameters

Figures 3.4.a and 3.4.b show the root loci of the position loop at varying K_{cp} , obtained with $T_{iv} = 1/|s_r|$ (low frequency zero-pole cancellation) and with two different choices of K_{cv} , respectively $K_{cv} = 5.2 \times 10^{-2}$, corresponding to a fast but poorly damped velocity loop, and $K_{cv} = 1.84 \times 10^{-2}$, corresponding to a slower but well damped loop. In the first case, two poles of the position closed loop are very near the process zeros, even for low values of K_{cp} , while in the second case the poles starting from the complex poles of the velocity closed loop go toward infinite. The value of K_{cv} discriminating this behaviour, in this example, has been found to be $K_{cv} = 2.4 \times 10^{-2}$, corresponding to $\omega_{cv} = 93.78$, i.e. $\omega_{cv} = 0.52\omega_p = 0.88\omega_z$, which resembles the well known rule of thumb for the choice of ω_{cv} , namely $\omega_{cv} \leq 0.5\omega_p$.

The time responses of q_m and nq_l to steps in setpoint and torque disturbances for both cases, computed with $K_{cp} = 30$ (i.e. approximately $\omega_{cp} = 30$) and without the feedforward derivative action, are given in Figures 3.5.a and 3.5.b respectively. The response to torque disturbances is dominated by the real pole canceled by the zero of the PI regulator. If the cancellation is avoided, this mode can be made faster. The disturbance rejection is stronger for $K_{cv} = 5.2 \times 10^{-2}$ but the load behaviour is more oscillatory.

3.3.2 On the Use of the Notch Filter. Commercial drives frequently enhance the P/PI control scheme using a notch filter, namely a filter with transfer function:

$$G_{nf}(s) = \mu \frac{s^2 + 2\xi_n\omega_n s + \omega_n^2}{s^2 + 2\xi_d\omega_d s + \omega_d^2}.$$

Sometimes the filter is inserted within the velocity loop, in series to the PI, sometimes outside. In principle, the filter can be used to avoid the excitation of oscillatory modes through the control variable. However, in industry there are controversial opinions about the effectiveness of the notch filter and on the best way to exploit it. Frequently, people rely on the notch filter to counteract oscillations which cannot get rid of otherwise. It is also suggested to use the notch filter to cancel the open loop complex poles and replace them with more damped poles [31]. Precise cancellation however is difficult because of uncertainty on the servo system parameters, especially the damping coefficient. Moreover, poles are canceled in the transfer functions involving the setpoint, but they remain unchanged in the closed loop and may be excited by other inputs, especially by torque disturbances.

Following the above root locus analysis, it is suggested in this contribution to place the filter outside the velocity loop and to select its zeros so as to

cancel the poles of the velocity loop. The position loop root locus, for the case $K_{cv} = 5.2 \times 10^{-2}$, becomes as in Figure 3.6.

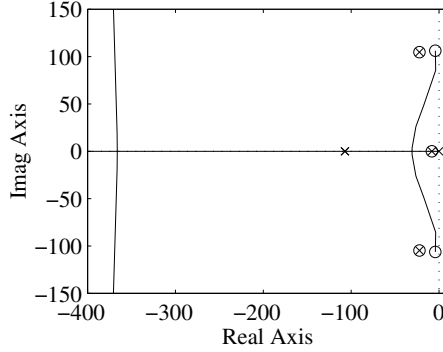


Figure 3.6. Root locus of the position loop with P/PI plus notch filter

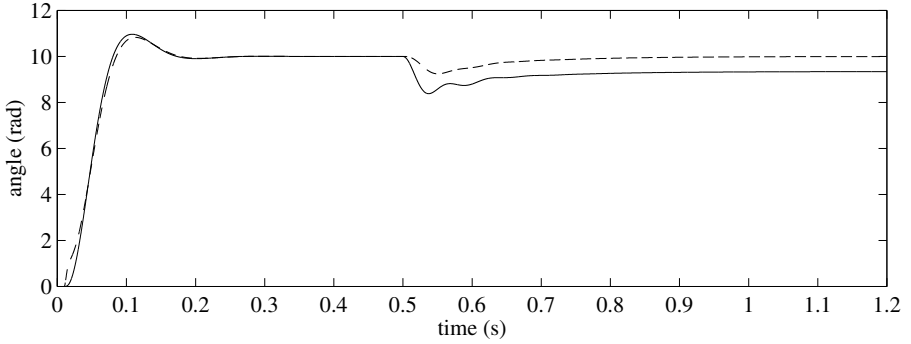


Figure 3.7. Time responses of q_m (dashed line) and nq_l (solid line) for $K_{cv} = 5.2 \times 10^{-2}$ with the notch filter

The closed loop dominant poles, again with $K_{cp} = 30$, are now well damped, even if the poorly damped poles of the velocity loop are still observable in the response to torque disturbances. As a result, the step responses of q_m and nq_l are as in Figure 3.7. There is a clear improvement in the response to the setpoint (the overshoot is due to the step input, which must be avoided) and a smaller one in that to the torque disturbance, whose dynamics is dominated by the canceled poles.

As a conclusion, designing jointly the velocity regulator and the notch filter, a good trade off can be obtained between velocity loop bandwidth and damping of the closed loop dominant poles. The velocity loop gain K_{cv} should be first chosen to get a desired damping of the velocity loop poles. Then, these poles are canceled (i.e. blocked) by the notch filter. Finally, the position loop gain K_{cp} is selected to achieve the desired damping of the dominant poles of the servo system.

The filter design proved also to be robust to errors in model parameters, since the notch filter zeros cancel out closed loop poles (the velocity loop ones), which are less sensitive to process parameter variations, than the open loop ones.

4 LQG Load Position Control

Considering the limitations of classical control with respect to the load behaviour, it is sensible to investigate solutions which take into account the whole state of the system under control, so as to completely assign the internal dynamics of the closed loop system, removing the lightly damped eigenvalues. The setpoint response of a particular state (the load position, in our context) will then be shaped through suitable feedforward actions.

Consider then the block diagram of Figure 4.1. The state-space plus

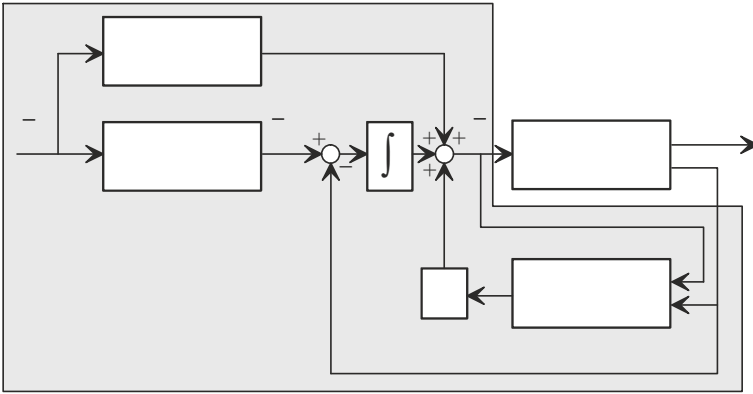


Figure 4.1. A state-space controller plus feedforward controller

feedforward controller is made up by the following functional blocks:

- Setpoint generator for the load position (not shown in figure).
- Setpoint conversion (from load to motor position).
- Feedforward for setpoint tracking.

- Integral term on the motor position error.
- Full order state observer or Kalman filter.
- State-space control law.

4.1 Design of the Optimal Controller

In state space form, model (3.1)–(3.4) is described as follows:

$$\dot{x} = Ax + Bu \quad (4.1)$$

$$y = Cx \quad (4.2)$$

with $x = [q_m \quad \dot{q}_m \quad q_l \quad \dot{q}_l]^T$, $B = [0 \quad K_t/J_m \quad 0 \quad 0]^T$, $u = \bar{I}$, $C = [1 \quad 0 \quad 0 \quad 0]^T$ (the motor position being the only measured variable), and:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{K_{el}}{J_m} & -\frac{D_m+D_{el}}{J_m} & \frac{nK_{el}}{J_m} & \frac{nD_{el}}{J_m} \\ 0 & 0 & 0 & 1 \\ \frac{nK_{el}}{J_l} & \frac{nD_{el}}{J_l} & -\frac{K_{el}}{J_{lr}} & -\frac{D_{el}}{J_{lr}} \end{bmatrix}.$$

The integral term on the motor position error is needed to eliminate steady state errors for constant disturbances, such as Coulomb friction. The gain k_I of the integral term is computed together with the matrix gain K of the state space control law, solving an optimal control problem. Denoting with x_I the state of the integral term, reference is thus made to an augmented state vector $\tilde{x} = [x^T \quad x_I]^T$, whose dynamics are given by:

$$\dot{\tilde{x}} = \tilde{A}\tilde{x} + \tilde{B}u + \tilde{G}\tilde{q}_m$$

with $\tilde{B} = [b^T \quad 0]^T$, $\tilde{G} = [0 \quad 0 \quad 0 \quad 0 \quad 1]^T$, while:

$$\tilde{A} = \begin{bmatrix} A & 0 \\ -c & 0 \end{bmatrix}.$$

As is well known, thanks to a separation principle [1], it is possible to carry out the design of the state space controller as if the whole state were available, postponing the problem of how to design a suitable dynamic system to get a correct estimation of the state. The quadratic loss function for the optimal LQ problem is:

$$J = \int_0^\infty [\tilde{x}(t)^T Q \tilde{x}(t) + u(t)^2] dt$$

The weight matrix Q is given the expression:

$$Q = \rho D D^T$$

where ρ is a scalar, while $D \in R^5$ is a vector to be determined in such a way that the following transfer function:

$$D^T (sI - \tilde{A})^{-1} \tilde{B} = \frac{m(s)}{p_0(s)}$$

(where $p_0(s)$ is the open loop characteristic polynomial of matrix \tilde{A}) has some desired properties. Specifically, it can be proven [1] that, if $m(s)$ is a fourth order polynomial, as ρ approaches infinity, four roots of the closed loop polynomial approach the roots of $m(s)$, while the remaining fifth root moves along the negative real axis away from the origin of the complex plane. The guideline to select the vector of weights D is thus to place the roots of polynomial $m(s)$ in desired positions of the complex plane (since four of the closed loop roots will approximately coincide with them). Then the scalar ρ is chosen as the minimum value needed to have the above asymptotic result hold approximately true. Values of ρ higher than necessary should be avoided, since they entail an excessive use of the control variable, as it is apparent from the expression of the loss function. Moreover, the crossover frequency of the control loop turns out to be proportional to $\sqrt{\rho}$.

As an example, for the prototype servo-positioning system described in Section 3.3.1, the roots of $m(s)$ have been chosen as in Figure 4.2 (two couples of complex roots, both with damping 0.7, one with natural frequency 80 rad/s, the other one with natural frequency 230 rad/s). Selecting $\rho = 10^5$, four of the closed loop eigenvalues are positioned on the roots of $m(s)$ while the remaining one is placed on the real axis in the point shown in Figure 4.2.

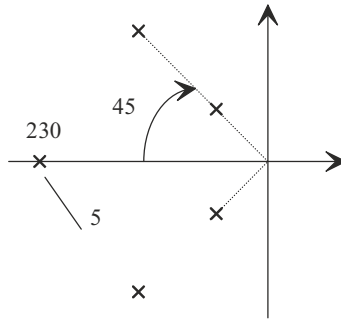


Figure 4.2. Roots of the polynomial $m(s)$

The selection of the weights in the loss function (the ever present problem of an optimal control design) has thus been converted to a somewhat partial pole placement. However, compared to standard pole placement design, the method offers as a byproduct a way to assess the validity of the choice made on the closed loop eigenvalues, given the open loop eigenvalues. Very high values of ρ are symptomatic of a bad choice, both in terms of control effort needed to impose the new desired dynamics to the system and in terms of

sensitivity of the closed loop eigenvalues to small errors in the parameters of the model. As an example, placing all the eigenvalues for our problem in the same point of the real axis (for example around -120 rad/s) needs a value of ρ three orders of magnitude higher and results in a dramatic increase of the sensitivity of the design to model uncertainties.

4.2 Design of the Kalman Filter

The design of the state observer has been carried out in a stochastic framework, using Kalman filter theory. Noise inputs have been added to the system model (4.1), (reffmrSS2) as follows:

$$\begin{aligned}\dot{x} &= Ax + B[u + v] \\ y &= Cx + w\end{aligned}$$

where v is a torque disturbance, while w is a disturbance on the measure. The noises are assumed to be uncorrelated, zero mean, white Gaussian noises, whose (scalar) variances are in fact all is needed for Kalman filter design.

As far as the variance of the noise on the output measure is concerned, the following formula has been used, which refers to the quantisation noise of the A/D converter:

$$\sigma_w^2 = \frac{h_c^2}{12} = \frac{1}{12} \left(\frac{\Delta_c}{2^{b-1}} \right)^2 = \frac{1}{3} 2^{-2b} \Delta_c^2$$

where h_c is the quantisation step of the converter, b is the number of bits used in the converter, Δ_c is the input range of the converter.

A similar formula could be used for computing the variance of the input disturbance, based on the quantisation of the D/A converter. However fine tuning of this value proved to be essential to get satisfactory results from the design of the filter.

4.3 Design of the Feedforward Filters

The last step in the design of the control system concerns the two open loop filters: the conversion of the load position setpoint to the motor position setpoint (this is needed to form the error which feeds the integrator) and the feedforward filter which shapes the setpoint response of the load. Let $G_r(s)$ and $G_f(s)$ be the transfer functions of the two filters, respectively (Figure 4.3).

The transfer function $G_r(s)$ should ideally be the inverse of the transfer function $G_{lm}(s)$.

On the other hand, the transfer function $G_f(s)$ should additionally invert the transfer function $G_k(s)$ of the system obtained feeding back the servo positioner with the state space control law, operating on the observed state (see

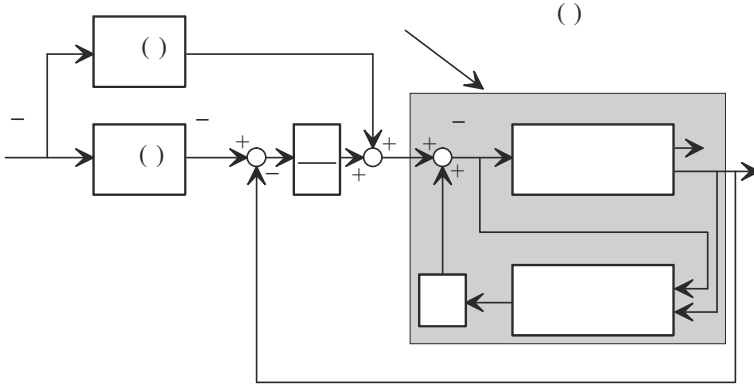


Figure 4.3. Design of the feedforward filters

Figure 4.3). It is straightforward to realise that the dynamics of the observer do not contribute to this transfer function, which takes the expression:

$$G_k(s) = C [sI - (A + BK)]^{-1} B$$

and its two zeros are the same as the zeros of the open loop transfer function (3.5) from \bar{I} to q_m .

In order to implement the two open loop filters with causal operators, the following choice is made:

$$G_r(s) = G_q(s)^{-1} F(s) \quad (4.3)$$

$$G_f(s) = G_k(s)^{-1} G_q(s)^{-1} F(s) \quad (4.4)$$

where $F(s)$ is a transfer function whose relative degree is at least equal to 3. Notice that this constraint is essential for the causality of (4.4). However introducing in (4.3) the same filter as in (4.4) allows concluding that the nominal transfer function from the load position setpoint to the load position is simply $F(s)$.

For example, $F(s)$ could be chosen as:

$$F(s) = \frac{1}{(1 + s/\omega_F^0) \left(1 + 2\zeta^0 s/\omega_n^0 + s^2/(\omega_n^0)^2 \right)}$$

For the same design of the control system characterised by the choice of the closed loop eigenvalues shown in Figure 4.2, ω_F^0 was set to 350 rad/s, ω_n^0 to 800 rad/s, ζ^0 to 0.7.

4.4 Experimental Results

The control scheme described above has been tuned and verified on the prototype servomechanism described in Section 3.3.1, and its performance has been assessed by comparison with a standard PID controller on the motor position error, tuned to get the maximum available bandwidth (about 80 rad/s).

Three experiments will be shown here, all of them characterised by a trapezoidal profile of the reference velocity, but with different parameters, as reported in Table 4.1. Figures 4.4 to 4.6 compare the load velocities with the two control schemes, obtained from an accelerometer placed on the load for validation purposes.

Experiment	1	2	3
Load rotation (deg)	40	30	50
Time for positioning (s)	0.5	0.5	0.9
Maximum acceleration (rad/s ²)	1200	900	500

Table 4.1. Parameters of the experiments

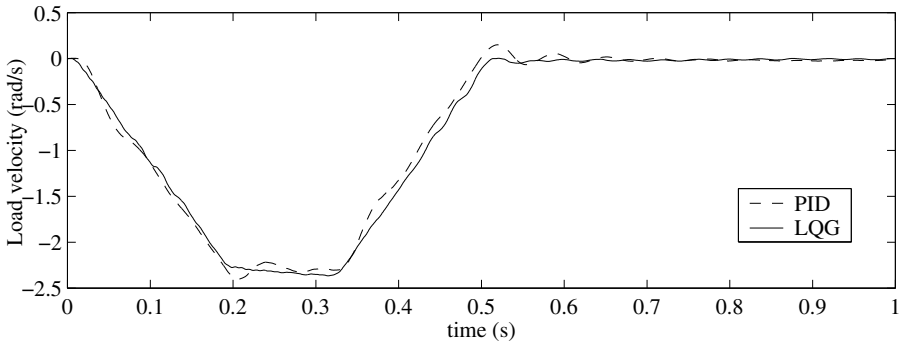


Figure 4.4. Load velocity in experiment 1

In the first experiment (where the fastest positioning has been commanded), the benefits of the LQG controller in terms of suppression of the load oscillations are evident: the large overshoot obtained with the PID controller almost disappears. Since this was the primary goal of the state-space controller, it can be concluded that the result of the implementation of the LQG has been successful. On the other hand, experiment 2 and, above all, experiment 3 (the slowest positioning) exhibit high frequency oscillations of the load with the LQG controller. These have been ascribed to noises, whose

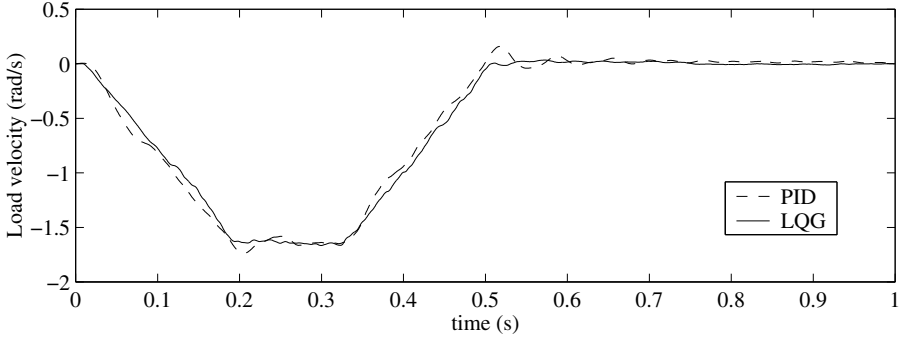


Figure 4.5. Load velocity in experiment 2

frequencies are proportional to the motor velocity, such as the torque ripple on the brushless motor, discussed in Section 2.3, and the ripple on the motor position transducer [11]. These noises enter the Kalman filter, where they are amplified, and their effect is visible on the observed load velocity.

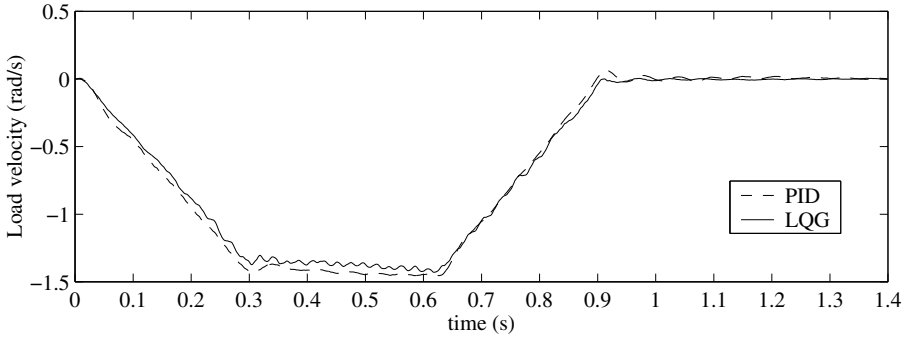


Figure 4.6. Load velocity in experiment 3

5 Conclusions and Future Directions

Modelling and control design of an elastic servo positioning drive and modelling, identification and compensation of the pulsating torque disturbances in PMAC BLM have been discussed in this paper. By feeding back only the motor position it is relatively easy to obtain satisfactory control of the motor position and velocity. Nevertheless this does not guarantee a satisfactory

control of the load position and velocity for demanding motion control applications. This happens especially during slow motion, when the frequencies of the pulsating disturbances generated by the motor and by the gear reducer match, and thus the servo structural resonance is excited.

It is shown how to design the widely used P/PI control to achieve a balanced trade off between a fast motor response and a damped behaviour of the load position. This helps matching the strong requirement of a smooth, non oscillating, slow velocity motion, typical of robotics, machine tools and other applications, but also points out the limits of P/PI control. P/PI control happens to be rather robust with respect to system parameter variations but its performance may not be fully satisfactory in those applications where the load velocity ripple is a major problem. To this purpose a new way to exploit a notch filter has been proposed, not only to prevent from harmonic signals (at the notch frequency) to enter the velocity loop but also to improve the damping of the closed loop (dominant) poles.

Modern control theory has been then used to design a LQG plus feed-forward controller: advantages with respect to classical control seem to be achievable for fast positioning manoeuvres, but the increased complexity of the implementation makes this choice still questionable for industrial use.

In addition to this, the reduction of the torque pulsation at source level is the most promising approach to the suppression of load oscillation at low velocity, to our experience. Therefore a compact model of pulsating torque disturbances suitable for online compensation has been given, together with algorithms for offline and online identification of the motor parameters. Experiments have been given which show the effectiveness of this approach for disturbance compensation.

References

1. Anderson B, Moore J 1989 *Optimal Control — Linear Quadratic Methods* Prentice Hall, London
2. Armstrong-Hélouvry B, Dupont P, Canudas de Wit C 1994 A survey of models, analysis tools and compensation methods for the control of machines with friction. *Automatica* 30:1083–1138
3. Clenet S, Lefèvre Y, Sadowski N, Astier S, Lajoie-Mazenc M 1993 Compensation of permanent magnet torque ripple by means of current supply waveshapes control determined by finite element method. *IEEE Transactions on Magnetics* 29:2019–2023
4. Favre E, Cardoletti L, Jufer M 1993 Permanent-magnet synchronous motors: A comprehensive approach to cogging torque suppression. *IEEE Transactions on Industrial Applications* 29:1141–1149
5. Ferretti G, Magnani G, Rocco P 1998 Modelling, identification and compensation of pulsating torque in permanent magnet AC motors. *IEEE Transactions on Industrial Electronics* 46:912–920

6. Ferretti G, Magnani G, Rocco P 1998 Online identification and compensation of torque disturbances in permanent magnet AC motors. In: *Proceedings of 1998 IEEE Industrial Electronics Conference* Aachen, D, pp 1521–1526
7. Ferretti G, Magnani G, Rocco P 1999 Force oscillations in contact motion of industrial robots: An experimental investigation. *IEEE/ASME Transactions on Mechatronics* 4:86–91
8. Forrest-Barlach M G, Babcock S M 1986 Inverse dynamics position control of a compliant manipulator. In: *Proceedings of 1986 IEEE International Conference on Robotics and Automation* San Francisco, CA, pp 196–205
9. Galbiati A 1998 *Online Estimation and Compensation of Torque Disturbances in a Brushless Motor* (in Italian) MS Thesis, Politecnico di Milano
10. Godler I, Ohnishi K, Yamashita T 1994 Repetitive control to reduce speed ripple caused by strain wave gearing. In: *Proceedings of 1994 IEEE Industrial Electronics Conference* Bologna, I, pp 1034–1038
11. Hanselman D 1991 Techniques for improving resolver-to-digital conversion accuracy. *IEEE Transactions on Industrial Electronics* 38:501–504
12. Hanselman D C 1994 Minimum torque ripple, maximum efficiency excitation of brushless permanent magnet motors. *IEEE Transactions on Industrial Electronics* 41:292–300
13. Holtz J, Springob L 1996 Identification and compensation of torque ripple in high-precision permanent magnet motor drives. *IEEE Transactions on Industrial Electronics* 43:309–320
14. Hung J Y, Ding Z 1993 Design of currents to reduce torque ripple in brushless permanent magnet motors. *IEE Proceedings – Part B* 140:260–266
15. Lee H S, Tomizuka M 1996 Robust motion controller design for high-accuracy positioning systems. *IEEE Transactions on Industrial Electronics* 43:48–55
16. Le-Huy H, Perret R, Feuillet R 1986 Minimization of torque ripple in brushless dc motor drives. *IEEE Transactions on Industrial Applications* 22:748–755
17. Lorenz R, Van Patten K 1991 High-resolution velocity estimation for all-digital AC servo drives. *IEEE Transactions on Industrial Applications* 27:701–705
18. Lozano R, Brogliato B 1992 Adaptive control of robot manipulators with flexible joints. *IEEE Transactions on Automatic Control* 37:174–181
19. Jahns T M 1984 Torque production in permanent-magnet synchronous motor drives with rectangular current excitation. *IEEE Transactions on Industrial Application* 20:803–813
20. Jahns T M, Soong W L 1996 Pulsating torque minimization techniques for permanent magnet ac motor drives – A review. *IEEE Transactions on Industrial Electronics* 43:321–330
21. Ji J K, Sul S K 1995 Kalman filter and LQ based speed controller for torsional vibration suppression in a 2-mass motor drive system. *IEEE Transactions on Industrial Electronics* 42:564–571
22. Marino R, Spong M W 1986 Nonlinear control techniques for flexible joint manipulators: a single link case study. In: *Proceedings of 1986 IEEE International Conference on Robotics and Automation* San Francisco, CA, pp 1030–1036
23. Nicosia S, Tomei P 1991 A PD control for trajectory tracking of flexible joint robots. In: *Proceedings of 3rd IFAC Symposium on Robot Control* Vienna, A, pp 255–260
24. Ohnishi K, Murakami T 1989 Application of advanced control techniques in electrical drives. In: *Proceedings of 1989 Workshop on Microcomputer Control of Electrical Drives* Trieste, I, pp 1–24
25. Ohnishi K, Shibata M, Murakami T 1996 Motion control for advanced mechatronics. *IEEE/ASME Transactions on Mechatronics* 1:56–67

26. Schmidt P B, Lorenz R 1992 Design principles and implementation of acceleration feedback to improve performance of DC drives. *IEEE Transactions on Industrial Electronics* 28:595–599
27. Spong M 1987 Modeling and control of elastic joint robots. *ASME Journal of Dynamic Systems, Measurement and Control* 109:310–319
28. Tomei P 1990 An observer for flexible joint robots. *IEEE Transactions on Automatic Control* 36:739–743
29. Umeno T, Hori Y 1991 Robust speed control of DC servomotors using modern two degrees of freedom controller design. *IEEE Transactions on Industrial Electronics* 38:363–368
30. Umeno T, Kaneko T, Hori Y 1993 Robust servosystem design with two degrees of freedom and its application to novel motion control of robot manipulators. *IEEE Transactions on Industrial Electronics* 40:473–485
31. Vukosavic S N, Stojic R 1998 Suppression of torsional oscillations in a high-performancs speed servo drive. *IEEE Transactions on Industrial Electronics* 45:108–117
32. Yao B, Al-Majed M, Tomizuka M 1997 High performance robust motion control of machine tools: an adaptive robust control approach and comparative experiments. *IEEE/ASME Transactions on Mechatronics* 2:63–76

Robotic Grasping and Manipulation

Antonio Bicchi¹ and Vijay Kumar²

¹ Centro “E. Piaggio”

Università degli Studi di Pisa

<http://www.piaggio.ccii.unipi.it/robotics-eng.html>

² Grasp Laboratory

University of Pennsylvania

In this chapter, we consider problems that arise in designing, building, planning, and controlling operations of robotic hands and end-effectors. The purpose of such devices is often manifold, and it typically includes grasping and fine manipulation of objects in an accurate, delicate yet firm way. We survey the state-of-the-art reached by scientific research and literature about the problems engendered by these often conflicting requirements, and the work that has been done in this area over the last two decades. Because of space limitations, the chapter does not attempt at providing a survey of the technology of robot hands, but rather it is oriented towards covering the theoretical framework, analytical results, and open problems in robotic manipulation.

1 Introduction

In many roboticists, the admiration for what nature accomplishes in everyday's functions of human beings and animals is the original stimulus for their research in emulating these capabilities in artificial life. Among the many awesome realisations of nature, few of the human abilities distinguish man from animals as deeply as manipulation and speech. Indeed, there are animals that can see, hear, walk, swim, etc. more efficiently than men — but language and manipulation skills are peculiar of our race, and constitute a continuing source of amazement for scientists. In this chapter, we will consider in detail the implementation of artificial systems to replicate in part the manipulating ability of the human hand.

The three most important functions of the human hand are to explore, to restrain, and to precisely move objects. The first function falls within the realm of *haptics*, an active research area in its own merits [48]. We will not attempt an exhaustive coverage of this area. The work in robot hands has mostly tried to understand and to emulate the other two functions. We will distinguish between the task of restraining objects, sometimes called *grasping* or *fixturing*, and the task of manipulating objects with fingers (in contrast to manipulation with the robot arm), sometimes called *dexterous manipulation*.

While grippers and fixtures have been used extensively in industry, one can argue that the field of robot grasping started with the work of Asada and Hanafusa [4] and Salisbury's first three-fingered robotic hand [61]. Since then,

many hand designs have been proposed, ranging from rather simple devices to very sophisticated multifingered hands such as the Utah-MIT hand [41]. Extensive surveys on robot hand systems are for instance those reported in [31], [37], [71], [94], and more recently [1], [75], [7].

In robot hand design, it can be observed that there are two prevailing philosophies, which can be identified with an *anthropomorphic* vs. a *minimalistic* approach to design. While the former philosophy basically attempts at replicating the human hand capabilities by imitating its mechanical structure, the latter focuses on realisation of some desirable grasping or manipulation features by purposeful design of mechanisms that have no intentional resemblance with any biological system. In the latter group, there have been a number of efforts focussing on reduced-complexity multifingered hands. Two examples of robot hands inspired to the two approaches and developed by groups participating in the RAMSETE project are reported in Figures 1.1 [64] and 1.2, respectively.



Figure 1.1. The University of Bologna dexterous hand

Design of robot hands still poses many challenges to the research community, and several are common to the two approaches above. However, it seems fair (though perhaps slightly oversimplifying) to affirm that anthropomorphic design is mostly confronted with technological problems such as accuracy and miniaturisation of sensors and actuators, power and signal transmission, etc. In minimalistic design, instead, the emphasis of current research is more on the theoretical analysis of manipulation systems, and their deep understanding in order to allow full exploitation of limited hardware capabilities. This chapter is more focussed on the latter class of problems.

Hardware complexity reduction can be achieved in several ways. For instance, when grasp robustness is considered, it can be observed that envelop-

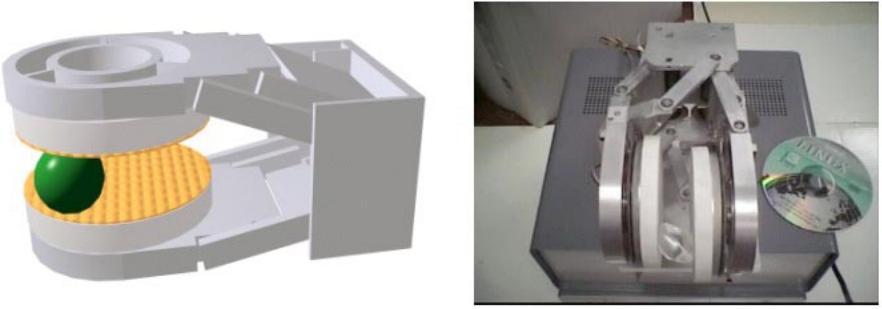


Figure 1.2. The University of Pisa dexterous gripper

ing grasps are superior in terms of restraining objects. Enveloping grasps [102], in contrast to fingertip grasps, are formed by wrapping the fingers (and the palm) around the object. Indeed, this is easily seen also in human grasping, where fingertips and distal phalanges are used in fingertip grasps for fine manipulation, while the inner parts of the hand (palm and proximal phalanges) are used in enveloping grasps for restraint [20], [40]). One of the first attempts at realising a reduced-complexity gripper was a three fingered hand powered by four actuators [104] that was designed to grasp by enveloping. Variations of this basic theme are also seen in grippers designed for the so-called *whole arm grasps* [89] and *power grasps* [67]. On the other hand, for achieving dexterous manipulation with a simplified hardware, the purposeful introduction of nonholonomic phenomena in manipulation by rolling has been advocated, and experimentally demonstrated, by several authors (see e.g. [19], [53], [71], [10]). Different modalities of manipulation and grasping share some fundamental theoretical framework, analytical results, and open problems, that are the subject of this chapter's survey.

2 Kinematics of Manipulation

The model of the hand we assume is comprised of an arbitrary number of fingers (i.e. simple chains of links —*phalanges*— connected through rotoidal or prismatic joints), and of an object, which is in contact with some or all of the phalanges. We let \mathbf{q} denote a vector of generalised coordinates, completely describing the configuration of the fingers; and $\mathbf{u} = (\mathbf{p}_o, \mathbf{R}_o) \in SE(3)$ denote the configuration (position and orientation) of the object. With a slight abuse

of notation, we also denote with $\dot{\mathbf{q}}$ and $\dot{\mathbf{u}}$ the elements of the tangent space to these configuration spaces (hence $\dot{\mathbf{u}} \in se(3)$ is the object twist).

Contacts represent a particular kind of kinematic constraint on the allowable configurations of the system, and cause most of the differences in the analysis of dexterous manipulation from other robotic systems. Contact constraints are typically unilateral, non-holonomic constraints on the generalised coordinates system, written in general in the form

$$\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{u}, \dot{\mathbf{u}}) \geq 0. \quad (2.1)$$

The inequality relationship reflects the fact that contact can be lost if the contacting bodies are brought away from each other. This involves an abrupt change of the structure of the model under consideration. To avoid analytical difficulties, it is usually assumed that manipulation is studied during time intervals when constraints hold with the equal sign (this is not the case in the study of grasping, where the study of these inequalities is crucial to understanding closure properties). The constraint relationship (2.1) is not in general integrable, i.e. it cannot be expressed in terms of \mathbf{q} and \mathbf{u} only: integrable constraints are called “holonomic”. Holonomic constraints between generalised coordinates reduce the number of independent coordinates necessary to describe the system configuration (degrees of freedom), and can be assumed to be removed from the description of the system by proper coordinate substitution. Nonholonomic constraints, on the contrary, do not reduce the number of degrees of freedom of the system, but rather reduce the number of independent coordinate velocities.

Contact kinematics is a study of the relationship between the location of the point of contact as a function of the relative motion of two contacting bodies. The first fundamental work in this area is due to Cai and Roth [16], who studied rigid planar bodies in point contact. They derived a relationship for the rates of change of the location of the point of contact as a function of the angular and linear velocities and accelerations of the contacting bodies. Montana [69] provided a more formal description of the configuration space associated with two contacting bodies, and derived the equations of kinematic contact that relate the time derivatives of contact coordinates with the relative angular and linear velocities. These equations include terms that depend on the curvature of the contacting bodies. Sarkar, Kumar, and Yun [90] extended this work to include acceleration terms. By using intrinsic geometric properties for the contacting surfaces, they showed the explicit dependence on the Christoffel symbols and their time derivatives. This set of results is directly relevant to dexterous manipulation [75], to the analysis of higher order closure properties [86], to stability analysis [36], and to manipulability by rolling [58].

To describe in more detail contact constraints that are in effect in dexterous manipulation systems, consider a contact between the i -th phalanx and the object, occurring at time t at a point described in an inertial base

frame B by the vector \mathbf{x}_i . A generic point on the surface of the phalanx will be described, in a frame C_i fixed on the phalanx, by the vector ${}^f\mathbf{x}_i$. Note that, ${}^f\mathbf{x}_i \in \mathbb{R}^3$ is actually bounded to lie on the surface S_i (which is assumed regular) of the link, and therefore can be regarded as a mapping ${}^f\mathbf{x}_i : {}^f\alpha_i \in U_i \subset \mathbb{R}^2 \mapsto S_i \subset \mathbb{R}^3$. The pair $(U_i, {}^f\mathbf{x}_i({}^f\alpha_i))$ is called a *chart* for (a portion of) the surface S_i , and the 2-vector ${}^f\alpha_i$ is referred to as the point coordinates on the i -th link. Orthogonal coordinates can be chosen so that the associated metric tensor is diagonal. A normalised Gauss frame can be associated with each point on the surface chart that has the origin in the point and is fixed w.r.t to the body so that its ζ axis is aligned with the outward pointing normal, while the χ and ξ axes span the tangent space. The orientation of the Gauss frame centered in \mathbf{x}_i w.r.t the C_i frame can be expressed by a rotation matrix ${}^f\mathbf{R}_i$. Similar considerations and definitions hold for the object surface.

Several types of contact models can be used to describe the interaction between the links and the object, among which the most common are the point-contact-with-friction model (or “hard-finger”), the “soft-finger” model, and the complete-constraint model (or “very-soft-finger”). In each case, the constraints consist in imposing that some components of the relative velocity between the Gauss frames that are associated with the contact point on each surface, are zero:

$$\mathbf{H}_i ({}^o\dot{\mathbf{c}}_i - {}^f\dot{\mathbf{c}}_i) = 0 \quad (2.2)$$

where \mathbf{H}_i is a constant selection matrix. Being the two frames fixed on the object and the phalanx, respectively, their velocities can be expressed as a function of the velocities of the object and of the joints as

$$\begin{aligned} {}^o\dot{\mathbf{c}}_i &= \mathbf{G}_i^T({}^o\alpha_i, \mathbf{u}) \dot{\mathbf{u}} \\ {}^f\dot{\mathbf{c}}_i &= \mathbf{J}_i({}^f\alpha_i, \mathbf{q}_i) \dot{\mathbf{q}}. \end{aligned}$$

Similar relationships hold for each contact point, and a single equation can be built to represent all constraints by properly juxtaposing vectors and block matrices to obtain

$$\mathbf{H}\mathbf{G}^T\dot{\mathbf{u}} - \mathbf{H}\mathbf{J}\dot{\mathbf{q}} = \begin{bmatrix} \mathbf{H}\mathbf{G}^T & -\mathbf{H}\mathbf{J} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{u}} \\ \dot{\mathbf{q}} \end{bmatrix} = 0. \quad (2.3)$$

The matrix \mathbf{G} is usually termed as the “grasp matrix”, or “grip transform”, while \mathbf{J} is referred to as the hand Jacobian.

One of the goals of the kinematic analysis of manipulation systems is to explicit the relationships between joint positions and object positions. Equation (2.3) can be used to this purpose (see e.g. [9]). Indeed, from (2.3), it is clear that the vector $[\dot{\mathbf{u}}^T \ \dot{\mathbf{q}}^T]^T$ must belong to a certain linear space, and hence that there exist three vectors ν_1 , ν_2 , and ν_3 (whose dimensions vary with the problem at hand) such that every possible pair of object velocity $\dot{\mathbf{u}}$ and joint velocity $\dot{\mathbf{q}}$ that comply with the kinematic and contact constraints of the hand system can be written as

$$\begin{aligned}\dot{\mathbf{u}} &= \mathbf{U}_o \nu_1 + \mathbf{U}_p \nu_2 \\ \dot{\mathbf{q}} &= \mathbf{Q}_p \nu_2 + \mathbf{Q}_o \nu_3.\end{aligned}\tag{2.4}$$

The columns of \mathbf{U}_p and those of \mathbf{Q}_p form a basis of the subspaces of compatible object and joint velocities, respectively. Any object motion described by the coordinate vector ν_2 in the image of \mathbf{U}_p must correspond to a joint motion with the same coordinates in the basis \mathbf{Q}_p . The images of \mathbf{Q}_o and \mathbf{U}_o represent the subspaces of redundant joint velocities and under-actuated object velocities, respectively.

Note that the matrices appearing on the right hand side of (2.4) are functions of the position of the contact point on the surfaces. If the dependency between \mathbf{u} , \mathbf{q} and ${}^o\alpha$, ${}^f\alpha$ is made explicit via the kinematics of rolling (see e.g. [69], [58]), explicit expressions for the joint motions that are required to perform a desired object motion can be obtained in principle. Notice also that, besides the analytical difficulties, in practice we often have the case that the geometry of the object is poorly known, if at all. The availability of contact sensors that are able to provide information on the position of the contact points on the phalanges is therefore necessary to attempt closed loop control of fine manipulation. In particular, if joint angles and contact points are sensed, (2.4) can be used even without information on the geometry of surfaces to control the object motion about desired trajectories by using generalised resolved-rate control.

3 Grasp Closure Properties

In order to define what grasping robustness is, the notions of *form closure* and *force closure* of a grasp are instrumental. These properties, first introduced by [84], concern the capability of the grasp to completely or partially constrain the motions of the manipulated object, and to apply arbitrary contact forces on the object itself, without violating friction constraints at the contacts.

3.1 Form Closure

Form-closure is the ability of a hand to prevent motions of the object, relying only on unilateral contact constraints. A mathematical definition of the problem can be stated as follows

Definition 3.1. *A configuration \mathbf{u}_0 of an object is form-closed by a hand in configuration \mathbf{q} if \mathbf{u}_0 is an isolated solution of the contact inequalities (2.1), i.e. if for all \mathbf{u} close to \mathbf{u}_0 , $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{u}, \dot{\mathbf{u}}) \geq 0 \Rightarrow \mathbf{u} = \mathbf{u}_0$.*

This purely geometric, rather general definition of form closure can be specialised to allow easy-to-check tests. In particular, in many cases it will suffice to look at the first-order approximation of the contact inequalities, which under rather general circumstances can be written as

$$\mathbf{N}^T \mathbf{G}^T \dot{\mathbf{u}} \geq 0 \quad (3.1)$$

where \mathbf{G} is the grasp matrix (evaluated at the current configuration) and \mathbf{N} is a matrix stacking contact normal vectors in its diagonal. Hence we have the following cases

- i) if there exists $\dot{\mathbf{u}}$ such that all components of $\mathbf{N}^T \mathbf{G}^T \dot{\mathbf{u}}$ are positive, the grasp is not form-closed;
- ii) if for all $\dot{\mathbf{u}}$, $\mathbf{N}^T \mathbf{G}^T \dot{\mathbf{u}}$ has at least one strictly negative component, the grasp is form closed;
- iii) if case i) does not apply, but there exists $\dot{\mathbf{u}}$ such that $\mathbf{N}^T \mathbf{G}^T \dot{\mathbf{u}}$ is nonnegative, the grasp may or may not be form-closed.

In cases i) and ii), second order terms are negligible, and form closure can be decided by first order arguments, using for instance linear programming. Specifically, case ii) is termed “first-order form closure”, and it corresponds to the most widely studied case in the literature. On the other hand, second or higher order effects must be taken into account in case iii).

First-order form closure (which also has direct bearing to the design of mechanical fixtures and jigs for manufacturing parts) has been studied since the 19th century. Early results showed that at least four frictionless contacts are necessary for grasping an object in the plane, and seven in the 3D case. In [68] and [59], it was shown that four and seven contacts are necessary and sufficient for the form-closure grasp of any polyhedron in the 2D and 3D case, respectively. An active area of research is the *synthesis* of form-closure grasps, i.e. given the object geometry, where to place contacts so as to prevent object motions. Constructive procedures for placing contacts on given objects to achieve form closure have attracted much attention in the literature, due also to the relevance to the fixturing problem (see e.g. the early work of [60], and more recently [33], [96], [11], [56], [55], [105]). There is also a form-closure *analysis* problem, i.e. given an object and a set of contact locations, to decide whether the object has any degree-of-freedom left, and which. Both qualitative (true/false) tests (see e.g. [51], [61], [68], [34]) and quantitative (quality index) tests [47], [101], [65] have been proposed for form closure. The extension of the classical, first-order notion of form closure to the so-called immobilisation problem, where second-order effects due to the relative curvature of the surfaces in contact are taken into account, has been introduced rather recently to provide more detailed results (see e.g. [35], [86], [103]) in case iii) above.

3.2 Force Closure

The analysis of form closure is intrinsically geometric, and does not take into account the kinematics and characteristics of the end-effector. While there is a wide consensus in the literature on the definition of form closure, the concept of force closure is somewhat less clear-cut and universally accepted.

The intuitive meaning of force closure implies that motions of the grasped object are completely (or partially) restrained despite whatever external disturbance, by virtue of suitably large contact forces that the constraining device (the end effector) is actually capable to exert on the object.

The force and moment balance equations for an object subject to an external force \mathbf{f} and moment \mathbf{m} , while grasped by a robotic mechanism by means of n contact forces \mathbf{p}_i applied at contact points \mathbf{c}_i , is written as

$$\mathbf{w} = \mathbf{G}\mathbf{p}, \quad (3.2)$$

where $\mathbf{w} = [\mathbf{f}^T \quad \mathbf{m}^T]^T$ is the external wrench, and $\mathbf{p} = [\mathbf{p}_1^T \quad \dots \quad \mathbf{p}_n^T]^T$. The relationship between contact forces and the torques at the m joints of the robotic hand can be written using the hand Jacobian as

$$\boldsymbol{\tau} = \mathbf{J}^T \mathbf{p},$$

A general solution of (3.2) can be written in the hypothesis that \mathbf{w} is *resistible* (i.e. $\text{rank } \mathbf{G} = \text{rank } [\mathbf{G} \quad \mathbf{w}]$) as

$$\mathbf{p} = \mathbf{G}^R \mathbf{w} + \mathbf{A}\mathbf{x}, \quad (3.3)$$

i.e. the sum of a particular solution of (3.2) (\mathbf{G}^R is a right-inverse of \mathbf{G}), and a homogeneous solution. \mathbf{A} is a matrix whose column form a basis of the null-space of \mathbf{G} . The coefficient vector $\mathbf{x} \in \mathbb{R}^{h_0}$ parametrises the homogeneous solution. Internal contact forces $\mathbf{p}_h = \mathbf{A}\mathbf{x}$ have no direct effect on the external wrench \mathbf{w} , but play an important role in the robustness of the equilibrium with respect to slippage induced by external disturbances, by allowing to “squeeze” the object in the grasp. It should be noted that, in general, for grasping mechanisms with few degrees of freedom, it may not be possible to apply arbitrary internal forces (see below Section 7).

In force-closure analysis one generally has to deal with frictional contacts. In different models of contact, such as the contact-point-with-friction, soft-finger, or very-soft-finger, friction forces and torques will be subject to limitations due to Coulomb’s law of friction or to its generalisations (see e.g. [30], [38]). We consider here contacts of the first type (generalisation poses no difficulties), for which Coulomb’s inequality holds,

$$\sigma_{i,f}(\mathbf{p}_i) = \alpha_i \|\mathbf{p}_i\| - \mathbf{p}_i^T \mathbf{n}_i < 0, \quad (3.4)$$

representing a cone in the space of contact forces \mathbf{p}_i . Substituting (3.3) in (3.4), an expression of friction constraints in terms of external wrenches and internal forces $\sigma_{i,f}(\mathbf{w}, \mathbf{x}) < 0$ is obtained. In these terms, we can state the following

Definition 3.2. *A grasp is defined Force-Closure if, for any external wrench \mathbf{w} acting on the object, there exists a vector \mathbf{x} such that all friction constraints are fulfilled.*

The analysis of force closure has been considered among others by [74], [28], [18], [73], while literature on the synthesis of force-closure grasps include [74], [79], [80], [81], [6].

According to the previous discussion on force closure, a crucial problem in robot manipulation is the choice of grasping forces so as to avoid (or minimise the risk of) slippage. The problem of choosing joint torques so as to realise the manipulating forces required by the task, while imposing internal forces that guarantee slippage avoidance, is often referred to as the *force distribution* problem. Further constraints on the choice of contact forces come from limitations in the object strength, or in the joint actuators torques. Accordingly, an “optimal” set of internal forces can be defined as the one that is further away from violating all such constraints. The force distribution problem is common with other robotic areas, as e.g. legged locomotion, cooperating and/or constrained manipulation, and has attracted much attention in the past few years (see e.g. [76], [46], [42], [54], [73], [106], [43], [78], [13]).

An important property of the nonlinear constrained optimisation problem to which grasp force distribution amounts is convexity. This property, used first in [6], enables efficient solutions to an otherwise very complex problem: [6] proposed numeric integration of an ODE as an iterative solution to the problem; [15] noticed that nonlinear friction constraints can be rewritten as positive-definiteness constraints on suitable matrices, and used projected gradient flow methods to optimise; [52] further exploited the matrix formulation of [15] to transform the problem in the format of a standard linear matrix inequality (LMI) problem, for which off-the-shelf, effective software exists.

4 Dynamics

The ability to predict the dynamic behaviour of a grasp with a given model including the control algorithms, is critical to the design of the grasp. In multifingered grippers, as in legged locomotion systems, multi-arm systems, and other constrained robot systems, several limbs are used to constrain and manipulate an object [50], [54], [67]. The dynamic analysis and the simulation (the prediction of motion given the external forces and moments on the system) of such systems is central to the design of such systems and the development of control algorithms [107], [91].

A hand-object system is a constrained mechanical system, whose dynamical description can be derived using Euler–Lagrange’s equations along with constraint equations. The disjoint dynamics of the hand and of the object are written as

$$\mathbf{M}_h(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{Q}_h(\mathbf{q}, \dot{\mathbf{q}}) = \boldsymbol{\tau}$$

$$\mathbf{M}_o(\mathbf{u})\ddot{\mathbf{u}} + \mathbf{Q}_o(\mathbf{u}, \dot{\mathbf{u}}) = \mathbf{w},$$

where $\mathbf{M}_h(\cdot)$ and $\mathbf{M}_o(\cdot)$ are symmetric positive definite composite inertia matrices, and $\mathbf{Q}_h(\cdot, \cdot)$ and $\mathbf{Q}_o(\cdot, \cdot)$ are terms including velocity-dependent and

gravity forces of the hand and of the object, respectively. Hand and object dynamics are linked through the n rigid-body contact constraints (2.1).

As we have seen before, when there are contacts between nominally rigid bodies, contact constraints are unilateral. Featherstone [27], Lotstedt [57] and Mason and Wang [62] pointed out some of the inconsistencies which arise when rigid body models are used with Coulomb’s empirical law of friction in unilateral systems. For example, if we consider the simulation of a rod sliding along a rough ground in a plane with a single contact, there are configurations in which no solutions (that are consistent with the constraints) exist, and others in which the solution is not unique. Wang, Kumar, and Abel [108], [107] performed a dynamic analysis of the peg-in-the-hole insertion problem and showed that there was a range of parameters during two-point-contact for which there were either no solutions or two solutions for the accelerations. Quasi-static analysis is also known to exhibit such inconsistencies [36].

The inconsistencies and ambiguities in the dynamic analysis of frictional contacts have been attributed to the approximate nature of Coulomb’s model and to the incorrect assumption of rigidity. Recently, there has been some attention in the robotics community on overcoming these shortcomings by using rigid body models to predict the gross motion while using compliant contact models to predict the contact forces and the local deformations [49].

One of the main difficulties that is present in multifingered grasps, and a feature that is particularly true of such grasps as power grasps and enveloping grasps, is that the number of independent contact forces is much larger than the number of actuators. Thus, from a controllability standpoint, not all the contact forces are controllable (see Section 7 below).

The analysis of statically indeterminate grasps or grasps in which there is no unique solution to the initial value problem is simply not possible unless one explicitly models the compliance at the contacts [20], [36], [74], [49]. Of course such contact models tend to be more complex and the parameters are more difficult to identify (see Section 6 below). Further, it is harder to simulate systems in which the time scale for the dynamics of contact interactions is significantly different from the time scale of rigid body dynamics [63], [98]. Thus, although efficient, approximate algorithms for “impulsive dynamic simulation” that incorporate approximate impact models for collisions are available [66], it is very difficult to write accurate simulators for dexterous and fine manipulation where the contact forces may be finite and the results may be sensitive to the parameters in the contact model.

5 Stability

A further important property of grasps is *stability*. The term is used in the literature with at least two meanings. One refers to Lyapunov theory, and dictates that a grasp is (asymptotically) stable if its dynamics are such that, when the object is displaced from its reference position, it will stay close (and

ultimately come back), to such position. A second definition is Lagrange's, whereby a grasp in which all forces are conservative, is stable if it corresponds to a strict local minimum of the potential energy. The second usage is prevalent in studies on grasp stability.

It is important to note that force closure does *not* guarantee stability. Any definition of stability must regard the grasp as a dynamic system and describe the properties of the dynamic system when it is perturbed from an equilibrium configuration. The role of compliance and dynamics in grasping has been investigated by many authors, beginning with Hanafusa and Asada [32] and Salisbury [61]. Cutkosky and Kao [21] discussed how to compute the aggregated compliance matrix of a hand-object system, including finger flexibility effects. Relations of compliant and rolling contacts with the stability of the grasp have been considered, at increasing levels of generality and detail, by [22], [70], [101], [36], [100], [29]. If Lagrange's stability criterion applies to an equilibrium grasp for a conservative system, Lyapunov stability follows. It should be noted however that Lagrange's analysis is limited under some regards. In mechanics, the seemingly intuitive statement that, if an equilibrium point is not a minimum for the potential function, then it is unstable, does not have a proof for systems with more than 2 degrees of freedom [3]. Perhaps more importantly, from an application viewpoint, is the fact that no provision is made in Lagrange analysis for non-conservative forces (except for Rayleigh-type dissipative terms). Nonconservative forces may arise in grasping systems because of nonidealities in the mechanical components, and of the control laws used for actuating the hand joints. The inclusion of the effects of control on the stability of grasp, which are apparently of major moment, is as of today a mostly open research problem. Lyapunov stability, and other structural properties (controllability, observability, stabilisability) of general grasping systems in their linear approximation have been investigated by [12], [82], [2]. Stable control of manipulation and grasping systems has been considered among others by [72], [85], [93], [87]. Particularly important is work done towards controlling grasping systems in the (practically ubiquitous) presence of uncertainties [17], [24].

A figure measuring stability (useful e.g. to compare different possible grasps) may be considered [36] as the real part of the dominant eigenvalue of the linearised grasp model (large values of this measure indicate that small perturbations are damped away quickly). An even more useful figure, in many applications, would be related to the size of the basin of attraction of the equilibrium, indicating how large a perturbation can be without causing instability: however, effective algorithms to evaluate such measure are not available at present.

6 Contact Compliance

The importance of modelling the finger-object contact and the role of compliance in grasping has been stressed by [4], [20], [95]. However, it is particularly difficult to model the relationship between small object/finger displacements and changes in contact forces arising from these displacements.

Such contact problems have been studied extensively in the solid mechanics community in the context of rail-wheel interaction [45] and analysis of ball and roller bearings [44]. There are difficulties even in establishing the uniqueness and existence of solutions of elastic bodies in static contact [25], and tractable analytical models are, in general, very difficult to come by. Hertz's model [44] can be used to predict the pressure distribution across each contact patch when the contacts are frictionless and non-conformal. Hertzian contact theory is probably the most widely used analytical contact model, and variations of this are used in [36], [86].

Because friction is central to robotic grasp, the Hertzian contact model has proved to be inadequate in many cases. Sinha and Abel [95] proposed an elastic contact stress model for finger-object contacts in multifingered grasping and a variational approach for quasi-static analysis. Wang, Kumar, and Abel [108] proposed a similar approach for dynamic analysis. They developed a mathematical programming approach for frictional, elastic contacts as well as viscoelastic contacts in which the inertial forces due to the deformations at the contacts are neglected. While such distributed parameter models yield accurate results, the solutions require computation-intensive numerical methods. A possible simplification is provided by the Winkler elastic foundation model [44], and the lumped parameter visco-elastic models used in [30], [49], [98] provide the simplest model for simulation and analysis.

One of the very hard problems is getting an accurate and tractable model of contact compliance, particularly in the tangential direction. This is recognised to be a difficult problem in the mechanics literature as well [44]. In addition to this, a tractable and accurate model of friction, one that accurately predicts slip and one that lends itself to stability analysis, is currently not available. Both these fundamental problem areas are crucial to robotic grasping and contact analysis.

For the purposes of analysis of grasp, it is generally assumed in the literature that all contacts are point contacts and idealisations such as a line or surface contact can be approximated by two or more point contacts. Each point contact can be modelled as either a frictionless point contact, a frictional point contact, or a soft contact [88]. A frictionless contact is defined as a contact in which the finger (or effector/fixture) can only exert a force along the common normal at the point of contact. A frictional contact (sometimes referred to as a point contact with friction) is defined as a contact that can transmit both a normal force and a tangential force, while a soft contact also allows the finger to exert a pure torsional moment about the common normal at the point of contact.

7 Grasping and the Kinematics of the Hand

It is interesting that much of the literature in grasping actually ignores the kinematics of the fingers or the articulations that are involved in contacting the object. While Reuleaux's problem of form closure justifiably focused on the geometry of the object and the arrangement of contacts, it is difficult to analyse a grasp without modelling the dynamics, or at least the kinematics, of the fingers and the interaction of the fingers with the object.

Trinkle *et al* explore the kinematics of enveloping grasps [102] using the restrictive but conservative assumption of frictionless contacts. The kinematics of fingers with two or three point contacts with fingertips and palms have been studied by [77], [26]. While the analysis of form closure is intrinsically geometric, force closure is tightly linked to the kinematics of the end effector. In fact, it is possible that a geometric analysis of a grasp may predict force closure, but a careful analysis of the kinematics may reveal that this is not the case [35]. Definitions of force closure that take into account the kinematics of the gripping device were proposed in [6], along with an exact algorithm for testing such property. Yoshikawa proposes a new set of definitions for closure properties, including what he calls active and passive closures, to explicitly model the properties of the grasping mechanism [110]. Unfortunately, much of this and other related work [39] is based on instantaneous kinematics.

Modelling of the fingers is particularly important when end effectors that have fewer degrees-of-freedom than necessary to impart arbitrary motions/forces at all contacts. Such *kinematically defective* grasps are common in simple industrial grippers. If the hand Jacobian matrix is not full rank, it is not possible to command an arbitrary set of grasp forces [5]. This is usually the case in all power grasps. Modelling of the kinematics and manipulability of whole-hand manipulation in such systems is discussed in [83]. Intuitively, the more a grasp is defective, the more robust it is in restraining an object with respect to external disturbances and the lower is sensitivity to positioning errors, but also the lower is manipulability. However, a case-by-case analysis is necessary for optimal power grasps [6].

Many open problems remain to be solved so as to design robot hands to effectively exploit defectivity to increase grasp robustness and reduce hardware complexity. Among these, perhaps the most important is the need for a reliable estimate of contact compliance, arising with statically indeterminate grasps. This will allow the calculation of contact forces, and the development of models that relate joint displacements and torques to contact forces.

8 Measures of Grasp Performance

Recent work in the literature has tried to develop quality measures for grasps. One such measure can be derived from the conditioning of the grasp matrix and is directly connected with the closure properties of the grasp [54]. In a

similar fashion, other structural properties can be derived from the characteristic matrices, for example, controllability and observability [83].

When an object is restrained or grasped with multiple effectors, there are two, often conflicting, measures of grasp performance. First, if the fixtures can be accurately positioned, the system's ability to reject wrench disturbances is a measure of grasp stability. The grasp stiffness matrix, or a frame invariant measure of the minimum grasp stiffness [14], provides one choice for a performance metric. This assumption of being able to accurately position the end effector is extensively used in the fixturing and grasping literature. However, when there are errors in positioning and orienting the end effectors, it is important to choose a grasp so that the system performance is insensitive to these positioning errors. Thus, it also makes sense to minimise the dependence of grasp forces on such positioning errors.

Howard and Kumar [36] develop the theory needed to combine the stiffness matrices at each contact to calculate a grasp stiffness matrix. While the signs of the eigenvalues allow a test of grasp stability, the eigenvalues themselves are not invariant with respect to changes in reference frames [35]. Bruyninckx *et al* [14] develops a frame invariant measure of stability that is based on the grasp stiffness matrix and a metric on the Euclidean group. Lin develops a frame-invariant quality measure that essentially minimises the "object deflection" when the grasped object is subject to force disturbances [55]. The basic idea here is to scale the eigenvalues measuring the rotational stiffness by a characteristic distance to an edge of the object. Thus it is possible to develop a scaled stiffness matrix and the smallest eigenvalue of the scaled matrix characterises the system.

The focus in the above work is to quantify the ability of a fixture to reject disturbances due to external forces on the workpiece [23]. This is clearly a measure of performance that is relevant. However, the robustness of a grasp to errors in positioning the effectors has not been addressed in this literature. Sugar and Kumar develop a second measure of performance that characterises this robustness and discuss an approach to optimising fixtures based on both measures [99]. In this connection, the control of grasping and the effects of uncertainties are particularly important.

Unfortunately most of these measures are based on the assumptions of small perturbations: displacements, forces and errors. It is no question that global measures would be more useful. For example, a figure relating to the size of the basin of attraction of the equilibrium, indicating how large a perturbation can be without causing instability would be desirable. However, the nonsmooth nature of grasp dynamics considerably complicates the analysis.

9 Concluding Remarks

This chapter has presented a survey of work in robotic grasping and manipulation over the last twenty years. It is impossible to do justice to all

the work in this area, particularly because of the breadth of the field and its close connection to dexterous manipulation, fixturing, and haptics. We chose to focus on issues that are central to the mechanics of grasping and the finger-object contact interactions. In addition, the review mainly addressed research that has established the theoretical framework for grasp analysis, simulation and synthesis. Because of the limitations on space, we have not given the algorithmic aspects, and the applications the attention that they deserve.

Acknowledgements

This paper builds upon previous works by the authors [7], [8]. The second author acknowledges support from NSF grants CISE RI 9703220, GRT 9355018, DARPA grants ITO/MARS 130-1303-4-534328-2000-0000 and ATO/TMR DAAH04-96-1-0007, and ARO grant MURI DAAH04-96-1-0007.

References

1. 2000 Goldberg K (Org) *Workshop on Part Feeding and Fixturing, 2000 IEEE International Conference on Robotics and Automation* San Francisco, CA
2. Al-Fahed Nuseirat A M, Hamdan A M A, Hamdan H M A 1999 Stability and modal control of an object grasped by a multifingered Robot Hand *Zeitschrift für Angewandte Mathematik und Mechanik* 79:473–479
3. Arnold V I 1980 *Mathematical Methods of Classical Mechanics* Springer, Heidelberg, D
4. Asada H 1979 *Studies on Prehension and Handling by Robot Hands with Elastic Fingers* PhD Thesis, Kyoto University
5. Bicchi A 1994 On the problem of decomposing grasp and manipulation forces in multiple whole-limb manipulation. *Journal of Robotics and Autonomous Systems* 13:127–147
6. Bicchi A 1995 On the closure properties of robotic grasping. *International Journal of Robotics Research* 14:319–334
7. Bicchi A 2000 Hands for dextrous manipulation and robust grasping: A difficult road towards simplicity. *IEEE Transactions on Robotics and Automation* 16:652–662
8. Bicchi A, Kumar V 2000 Robotic grasping and contact: A review, In: *Proceedings of 2000 IEEE International Conference on Robotics and Automation* San Francisco, CA, pp 348–353
9. Bicchi A, Melchiorri C, Balluchi D 1995 On the mobility and manipulability of general multiple limb robotic systems. *IEEE Transactions on Robotics and Automation* 11:215–228
10. Bicchi A, Sorrentino R 1995 Dexterous manipulation through rolling. In: *Proceedings of 1995 IEEE International Conference on Robotics and Automation* Nagoya, J, pp 452–457

11. Borst Ch, Fischer M, Hirzinger G 1999 A fast and robust grasp planner for arbitrary 3D objects. In: *Proceedings of 1999 IEEE International Conference on Robotics and Automation* Detroit, MI, pp 1890–1896
12. Brook N, Shoham M, Dayan J 1998 Controllability of grasps and manipulations in multi-fingered hands. *IEEE Transactions on Robotics and Automation* 14:185–192
13. Bruyninckx H, Demey S, Dutré S, DeSchutter J 1995 Kinematic models for model based compliant motion in the presence of uncertainty. *International Journal of Robotics Research* 14:465–482
14. Bruyninckx H, Demey S, Kumar V 1998 Generalized stability of compliant grasps. In: *Proceedings of 1998 IEEE International Conference on Robotics and Automation* Leuven, B, pp 2396–2402
15. Buss M, Hashimoto H, Moore J 1996 Dextrous hand grasping force optimization. *IEEE Transactions on Robotics and Automation* 12:406–418
16. Cai C, Roth B 1987 On the spatial motion of rigid bodies with point contact. In: *Proceedings of 1987 International Conference on Robotics and Automation* Rayleigh, NC, pp 686–695
17. Cheah C C, Han H-Y, Kawamura S, Arimoto S 1998 Grasping and position control for multi-fingered robot hands with uncertain Jacobian matrices. In: *Proceedings of 1998 IEEE International Conference on Robotics and Automation* Leuven, B, pp 2403–2408
18. Chen I-M, Burdick J W 1993 A qualitative test for n -finger force-closure grasps on planar objects with applications to manipulation and finger gaits. In: *Proceedings 1993 IEEE International Conference on Robotics and Automation* Atlanta, GA, pp 2:814–820
19. Cole A A, Hsu P, Sastry S S 1992 Dynamic control of sliding by robot hands for regrasping. *IEEE Transactions on Robotics and Automation* 8:42–52
20. Cutkosky M R 1985 *Robotic Grasping and Fine Manipulation* Kluwer, Boston, MA
21. Cutkosky M R, Kao I 1989 Computing and controlling the compliance of a robotic hand. *IEEE Transactions on Robotics and Automation* 5:151–165
22. Cutkosky M, Wright P 1986 Friction, stability and the design of robotic fingers. *International Journal of Robotics Research* 5:20–37
23. Donoghue J, Howard W S, Kumar V 1994 Stable workpiece fixturing. In: *Proceedings 3rd Biennial ASME Mechanisms Conference* Minneapolis, MN, pp 475–482
24. Doulgeri Z, Arimoto S 1999 A force control for a robot finger under kinematic uncertainties. In: *Proceedings of 1999 IEEE International Conference on Robotics and Automation* Detroit, MI, pp 1475–1480
25. Duvaut G, Lions J L 1976 *Inequality in Mechanics and Physics* Springer, London, UK
26. Erdmann M 1997 An exploration of nonprehensile two-palm manipulation. *International Journal of Robotics Research* 16:1–23
27. Featherstone R 1986 The dynamics of rigid body systems with multiple concurrent contacts. In: Faugeras O D, Giralt G (Eds) *Robotics Research: The Third International Symposium* MIT Press, Cambridge, MA, pp 189–196
28. Ferrari C, Canny J 1992 Planning optimal grasps. In: *Proceedings of 1992 IEEE International Conference on Robotics and Automation* Nice, F, pp 2290–2295
29. Funahashi Y, Yamada T, Tate M, Suzuki Y 1996 Grasp stability analysis considering the curvatures at contact points. In: *Proceedings of 1996 IEEE International Conference on Robotics and Automation* Minneapolis, MN, pp 3040–3046

30. Goyal S, Pinson E M, Sinden F W 1994 Simulation of dynamics of interacting rigid bodies including friction — I: General problem and contact model. *Engineering with Computers* 10:162–174
31. Grupen R A, Henderson T C, McCammon I D 1989 A survey of general-purpose manipulation. *International Journal of Robotics Research* 8:38–62
32. Hanafusa H, Asada H 1982 Stable prehension by a robot hand with elastic fingers. In: Brady M, Hollerbach J M, Johnson T L, Lozano-Pérez T, Mason M T (Eds) *Robot Motion Planning and Control* MIT Press, Cambridge, MA, pp 323–336
33. Hester R D, Cetin M, Kapoor C, Tesar D 1999 A criteria-based approach to grasp synthesis In: *Proceedings of 1999 IEEE International Conference on Robotics and Automation* Detroit, MI, pp 1255–1260
34. Hirai S, Asada H 1993 Kinematics and statics of manipulation using the theory of polyhedral convex cones. *International Journal of Robotics Research* 12:434–447
35. Howard W S 1995 *On the Stability of Grasped Objects* PhD Thesis, University of Pennsylvania
36. Howard W S, Kumar V 1996 On the stability of grasped objects. *IEEE Transactions on Robotics and Automation* 12:904–917
37. Howe R D, Cutkosky M R 1992 Touch sensing for robotic manipulation and recognition. In: Khatib O, Craig J J, Lozano-Pérez (Eds) *The Robotics Review 2* MIT Press, pp 55–112
38. Howe R D, Cutkosky M R 1996 Practical force-motion models for sliding manipulation. *International Journal of Robotics Research* 15:557–572
39. Hunt K H, Samuel A E, McAree P R 1991 Special configurations of multi-finger multi-freedom grippers — A kinematic study. *International Journal of Robotics Research* 10:123–134
40. Iberall T 1997 Human prehension and dexterous robot hands. *International Journal of Robotics Research* 16:285–299
41. Jacobsen S C, Knutti D F, Johnson R T, Sears H H 1982 Development of the Utah artificial arm. *IEEE Transactions on Biomedical Engineering* 29:249–269
42. Jameson J W 1985 *Analytic Techniques for Automated Grasp* PhD Thesis, Stanford University
43. Joh J, Lipkin H 1991 Lagrangian wrench distribution for cooperating robotic mechanisms. In: *Proceedings of 1991 IEEE International Conference on Robotics and Automation* Sacramento, CA, pp 224–229
44. Johnson K 1985 *Contact Mechanics* Cambridge University Press, Cambridge, UK
45. Kalker J J 1990 *Three-Dimensional Elastic Bodies in Rolling Contact* Kluwer, Boston, MA
46. Kerr J, Roth B 1986 Analysis of multifingered hands *International Journal of Robotics Research* 4:3–17
47. Kirkpatrick D, Kosaraju S R, Mishra B, Yap C–K 1989 *Quantitative Steinitz's Theorem with Applications to Multifingered Grasping* TR460, Courant Institute of Mathematical Sciences, New York University
48. Klatzky R, Lederman S 1990 Intelligent exploration by the human hand. In: Venkataraman S T, Iberall T (Eds) *Dextrous Robot Manipulation* Springer, London, UK
49. Kraus P R, Fredriksson A, Kumar V 1997 Modeling of frictional contacts for dynamic simulation. In: *Proceedings of Workshop on Dynamic Simulation: Methods and Applications, 1997 IEEE/RSJ International Symposium on Robotic and Intelligent Systems* Grenoble, F, pp 33–38

50. Kumar V, Waldron K J 1988 Force distribution in closed kinematic chains. *IEEE Journal of Robotics and Automation* 4:313–321
51. Lakshminarayana K 1978 *Mechanics of Form Closure* ASME Technical Report 78-DET32
52. Li H, Trinkle J C, Li Z 1999 Grasp analysis as linear matrix inequality problems. In: *Proceedings of 1999 IEEE International Conference on Robotics and Automation* Detroit, MI, pp 1261–1266
53. Li Z, Canny J 1990 Motion of two rigid bodies with rolling constraint. *IEEE Transactions on Robotics and Automation* 6:62–72
54. Li Z, Hsu P, Sastry S S 1989 Grasping and coordinated manipulation by a multifingered robot hand. *International Journal of Robotics Research* 8:42–52
55. Lin Q, Burdick J W 1999 A task-dependent approach to minimum-deflection fixtures. In: *Proceedings of 1999 IEEE International Conference on Robotics and Automation* Detroit, MI, pp 1562–1567
56. Liu Y H, Ding D, Wang S 1999 Constructing 3D frictional form-closure grasps of polyhedral objects. In: *Proceedings of 1999 IEEE International Conference on Robotics and Automation* Detroit, MI, pp 1904–1909
57. Lustedt P 1982 Mechanical systems of rigid bodies subject to unilateral constraints. *SIAM Journal of Applied Mathematics* 42:281–296
58. Marigo A, Bicchi A 2000 Rolling bodies with regular surfaces: Controllability theory and applications. *IEEE Transactions on Automatic Control* 45:1586–1599
59. Markenscoff X, Ni L, Papadimitriou C H 1990 The geometry of grasping. *International Journal of Robotics Research* 9:61–74
60. Markenscoff X, Papadimitriou C H 1989 Optimum grip of a polygon. *International Journal of Robotics Research* 8:17–29
61. Mason M T, Salisbury J K 1985 *Robot Hands and the Mechanics of Manipulation* MIT Press Cambridge, MA
62. Mason M T, Wang Y 1988 On the inconsistency of rigid-body frictional planar mechanics. In: *Proceedings of 1988 IEEE International Conference on Robotics and Automation* Philadelphia, PA, pp 524–528
63. McClamroch N H 1989 A singular perturbation approach to modeling and control of manipulators constrained by a stiff environment. In: *Proceedings of 28th IEEE Conference on Decision and Control* Tampa, FL, pp 2407–2411
64. Melchiorri C, Vassura G 1992 Mechanical and control features of the University of Bologna Hand Version 2. In: *Proceedings of 1992 IEEE/RSJ International Conference on Intelligent Robots and Systems* Raleigh, NC, pp 187–193
65. Miller A T, Allen P K 1999 Examples of 3D grasp quality computations. In: *Proceedings of 1999 IEEE International Conference on Robotics and Automation* Detroit, MI, pp 1240–1245
66. Mirtich B, Canny J 1996 Estimating pose statistics for robotic part feeders. In: *Proceedings of 1996 IEEE International Conference on Robotics and Automation* Minneapolis, MN, pp 1140–1146
67. Mirza K, Orin D E 1994 General formulation for force distribution in power grasp. In: *Proceedings of 1994 IEEE International Conference on Robotics and Automation* San Diego, CA, pp 880–887
68. Mishra B, Schwartz J T, Sharir M 1987 On the existence and synthesis of multifinger positive grips. *Algorithmica* 2:541–558
69. Montana D J 1988 The kinematics of contact and grasp. *International Journal of Robotics Research* 7:17–32
70. Montana D J 1992 Contact stability for two-fingered grasps. *IEEE Transactions on Robotics and Automation* 8:421–430

71. Murray R M, Li Z, Sastry S S 1994 *A Mathematical Introduction to Robotic Manipulation* CRC Press, Boca Raton, FL
72. Nagai K, Yoshikawa T 1993 Dynamic manipulation/grasping control of multifingered robot hands. In: *Proceedings of 1993 IEEE International Conference on Robotics and Automation* Atlanta, GA, pp 3:1027–1032
73. Nakamura Y, Nagai K, Yoshikawa T 1989 Dynamics and stability in coordination of multiple robotic systems. *International Journal of Robotics Research* 8:44–61
74. Nguyen V D 1988 Constructing force-closure grasps. *International Journal of Robotics Research* 7:3–16
75. Okamura A M, Smaby N, Cutkosky M R 2000 An overview of dextrous manipulation. In: *Proceedings of 2000 IEEE International Conference on Robotics and Automation* San Francisco, CA, pp 255–262
76. Orin D E, Oh S Y 1981 Control of force distribution in robotic mechanisms containing closed kinematic chains. *ASME Journal of Dynamic Systems, Measurement and Control* 102:134–141
77. Paljug E, Yun X, Kumar V 1994 Control of rolling contacts in two-arm manipulation. *IEEE Transactions on Robotics and Automation* 10:441–452
78. Panagiotopoulos P D, Al-Fahed A M 1994 Robot hand grasping and related problems: optimal control and identification. *International Journal of Robotics Research* 13:127–136
79. Park Y C, Starr G P 1992 Grasp synthesis of polygonal objects using a three-fingered robot hand. *International Journal of Robotics Research* 11:163–184
80. Ponce J, Faverjon B 1995 On computing three finger force-closure grasp of polygonal objects. *IEEE Transactions on Robotics and Automation* 11:868–881
81. Ponce J, Sullivan S, Sudsang A, Boissonnat J D, Merlet J P 1997 On computing four-finger equilibrium and force-closure grasps of polyhedral objects. *International Journal of Robotics Research* 16:11–35
82. Prattichizzo D, Bicchi A 1997 Consistent specification of manipulation tasks for defective mechanical systems *ASME Journal of Dynamical Systems, Measurement and Control* 119:767–775
83. Prattichizzo D, Bicchi A 1998 Dynamic analysis of mobility and graspability of general manipulation systems. *IEEE Transactions on Robotics and Automation* 14:241–258
84. Reuleaux F 1875 *Theoretische Kinematic*. Translated 1963 as *Kinematics of Machinery* Dover, New York, NY
85. Reznik D, Lumelsky V 1994 Multi-finger “hugging”: A robust approach to sensor based grasp planning. In: *Proceedings of 1994 IEEE International Conference on Robotics and Automation* San Diego, CA, pp 754–759
86. Rimón E, Burdick J W 1998 Mobility of bodies in contact. *IEEE Transactions on Robotics and Automation* 14:696–717
87. Rodriguez-Angeles A, Parra-Vega V 1988 Adaptive control with impedance of cooperative multi-robot system. In: *Proceedings of 1988 IEEE International Conference on Robotics and Automation* Philadelphia, PA, pp 1522–1527
88. Salisbury J K 1982 *Kinematics and Force Analysis of Articulated Hands* PhD Thesis, Stanford University
89. Salisbury J K 1987 Whole-arm manipulation. In: Bolles R, Roth B (Eds) *Robotics Research: The 4th International Symposium* MIT Press, Cambridge, MA, pp 183–189
90. Sarkar N, Kumar V, Yun X 1996 Velocity and acceleration equations for three-dimensional contact. *ASME Journal of Applied Mechanics* 63:974–984
91. Sarkar N, Yun Y, Kumar V 1997 Dynamic control of 3-D rolling contacts in two-arm manipulation. *IEEE Transactions on Robotics and Automation* 13:364–376

92. Selig A J, Rooney P K 1989 Reuleaux pairs and surfaces that cannot be gripped. *International Journal of Robotics Research* 8:79–87
93. Sheridan M J, Ahalt S C, Orin DE 1995 Fuzzy control for robotic power grasp. *Advanced Robotics* 9:535–546
94. Shimoga K B 1996 Robot grasp synthesis algorithms: A survey. *International Journal of Robotics Research* 15:230–266
95. Sinha P R, Abel J M 1992 A contact stress model for multifingered grasps of rough objects. *IEEE Transactions on Robotics and Automation* 8:7–22
96. Smith G, Lee E, Goldberg K Y, Bohringer K, Craig J 1999 Computing parallel-jaw grips. In: *Proceedings of 1999 IEEE International Conference on Robotics and Automation* Detroit, MI, pp 1897–1902
97. Somoff R 1897 Uber schraubengeschwindigkeiten eines festen korpers bei verschiedener zahl von stutzflachen, *Zietschrift für Mathematik und Physik* 42:133–153
98. Song P, Kraus P, Kumar V, Dupont P 1999 A singular perturbation analysis of the dynamics of systems with frictional contacts. *ASME Journal of Applied Mechanics* 67
99. Sugar T, Kumar V 1998 Decentralized control of cooperating mobile manipulators. In: *Proceedings of 1998 IEEE International Conference on Robotics and Automation* Leuven, B, pp 2916–2921
100. Svinin M M, Ueda K, Kaneko M 1999 Analytical conditions for the rotational stability of an object in multi-finger grasping. In: *Proceedings of 1999 IEEE International Conference on Robotics and Automation* Detroit, MI, pp 1347–1352
101. Trinkle J C 1992 On the stability and instantaneous velocity of grasped frictionless objects. *IEEE Transactions on Robotics and Automation* 8:560–572
102. Trinkle J C, Abel J M, Paul R P 1987 An investigation of frictionless enveloping grasping in the plane. *International Journal of Robotics Research* 7:33–51
103. Trinkle J C, Farahat A O, Stiller P F 1995 First order stability cells of active multi-rigid body systems. *IEEE Transactions on Robotics and Automation* 11:545–557
104. Ulrich N, Kumar V, Paul R, Bajcsy R 1990 Grasping with mechanical intelligence. In: *Preprints of CISM-IFToMM Symposium on Robots, Manipulators, and Systems* Kraków, PL, pp 333–339
105. van der Stappen A F, Wentink C, Overmars M H 1999 Computing form-closure configurations In: *Proceedings of 1999 IEEE International Conference on Robotics and Automation* Detroit, MI, pp 1837–1842
106. Walker I D, Freeman R A, Marcus S I 1991 Analysis of motion and internal loading of objects grasped by multiple cooperating manipulators. *International Journal of Robotics Research* 10:396–409
107. Wang Y, Kumar V 1994 Simulation of mechanical systems with unilateral constraints. *Journal of Mechanical Design* 116:571–580
108. Wang Y-T, Kumar V, Abel J 1992 Dynamics of rigid bodies with multiple frictional contacts. In: *Proceedings of 1992 IEEE International Conference on Robotics and Automation* Nice, F, pp 2764–2769
109. Wen J T, Wilfinger L S 1998 Kinematic manipulability of general constrained rigid multibody systems. In: *Proceedings of 1998 IEEE International Conference on Robotics and Automation* Leuven, B, pp 1020–1025
110. Yoshikawa T 1996 Passive and active closures by constraining mechanisms. In: *Proceedings of 1996 IEEE International Conference on Robotics and Automation* Minneapolis, MN, pp 1477–1484

Tactile Sensing for Robotic Manipulation

Claudio Melchiorri

LAR, Dipartimento di Elettronica Informatica e Sistemistica
Università degli Studi di Bologna
<http://www-lar.deis.unibo.it>

In several fields of robotics, tactile and force sensors represent a basic tool for achieving an enhanced interaction with the environment. As a matter of fact, areas such as advanced manipulation, telemanipulation, haptic devices, legged robots and so on are intrinsically based on an advanced sensorial equipment and on proper techniques for the exploitation of their information. These types of sensors give information such as the presence of a contact, its size and shape, the exchanged forces/torques. More advanced sensors can also provide additional information, such as mechanical properties of the bodies in contact (e.g. friction coefficient, roughness, ...) or the slippage. In this chapter, an overview on tactile and force sensors and their specific use in robotic manipulation is presented. In particular, after an illustration of the state of the art and of the main recent technological developments, results concerning the detection and control of the relative motion (slippage) of two bodies in contact are discussed. This problem may be of relevance, e.g. in advanced manipulation by robotic systems in which, depending on the task to be executed, it might be desirable either to avoid or to exploit the slippage of the manipulated object. Similar problems can be found with legged robots, or more in general in any case where a robotic device has to interact with its environment in a controlled manner. These and other applications have justified an increasing research effort in this field in the last years, generating several interesting prototypes and techniques for data analysis.

1 Introduction

In the last decades, the development of new technologies has allowed a growth of applications for industrial robots, which in turn has led to important achievements in the production field as well as in other sectors, such as medical care, support for disable people, help to human beings in hazardous environments. In this wide scenario, one of the main bottlenecks which still limits an even broader diffusion of robotic manipulation systems is the sensorial equipment of these devices. As a matter of fact, a typical industrial robot is equipped only with a relatively poor sensory capability, consisting of joint position sensors. Then, depending on the applications the robot may be equipped with additional sensorial devices, such as force/torque sensors, video cameras, or special sensors designed for the specific task, for example sensors for measuring temperature, pressure, humidity and so on.

The limited sensoriality of a robot manipulator has two main implications. On one hand several tasks are not achievable by current robots since they are not able to handle complex situations due to their limited knowledge of the environment. On the other hand, even when tasks are possible, the programming efforts (and therefore the time needed for the set-up phase) is much larger than in case of a sensorised manipulator. Another important consideration derives from the fact that in the last years new kinds of manipulation devices have been, and still are, developed both in the research and in the advanced application domain (space, underwater, medical, ...). These new devices can be generically considered as ‘advanced manipulation systems’, and may consist of multi-fingered robotic hands, cooperating robots, walking machines, and so on [34], [40], [9]. These manipulation devices requires as a basic need an advanced sensoriality, able to measure and monitor the interactions with manipulated objects, and more in general with the environment.

In robotic manipulation, two main families of sensors for measuring the variables of interest during interactions with the environment can be mentioned. The first category consists of sensors measuring properties of the environment without a direct interaction with it: vision systems, US-based sensors, proximity sensors, etc.. The second category includes sensors which provide a measurement of variables generated during the physical interaction between the robot and the environment. Typical examples are force and/or tactile sensors.

Undoubtedly, more attention has been devoted by the robotic and artificial intelligence communities to the first class of sensors. Reasons of this fact probably are the commercial availability of economic vision or US systems, and the fact that often an analysis of static information of the environment is sufficient for planning the tasks of the robotic device.

On the other hand, it has been recognised the importance of the availability of more precise data during the evolution of a manipulation task. Information such as friction, applied force, incipient slippage, thermal properties, and so on, cannot be obtained using a vision or US sensor. Moreover there may be factors, for example a poor illumination of the environment, which limit or do not permit at all the use of this type of sensorial devices. For these reasons, in the last decade a noticeable activity has been devoted to the development of tactile and force sensing devices.

In this chapter, problems relative to tactile/force sensing in robotic manipulation are considered. An overview on the state of the art and on the technologies adopted for the realisation of tactile and force sensors is given in Section 2. In Section 3 the use of force sensors as tactile (Intrinsic Tactile) devices is illustrated, as well as a tactile matrix sensor and its use for manipulation purposes. In Section 4, techniques for slip detection and control are described, reporting also an overview on recent results in this field. Results obtained for integrating the two sensors are illustrated in Section 5, for both detecting the slip and controlling it in order to execute some type

of manipulation. A key point here is that the integration of the two sensing devices allows obtaining new capabilities, not achievable with the single sensors alone. The conclusive section reports comments and future research directions.

2 An Overview on Force/Torque and Tactile Sensors

As previously mentioned, force and/or tactile sensing is one of the most basic requirements for any manipulator interacting physically with the environment in a non-conventional or non-structured manner. By means of direct measurements on the contacted objects, it is possible to apply control and behaviour strategies in order to accomplish complex tasks in autonomous or semi-autonomous way. Other sensory equipment can provide only generic information on manipulator's interaction with the environment. For example vision systems, that cannot give precise and local information on the contact state, are often used in the planning phase. For these reasons, the design of tactile/force sensors and the development of techniques to exploit their information are considered relevant research areas for the robotic community [32], [31], [26], [17], [25].

2.1 Force Sensors, IT Sensing

Commercially available force sensors are devices to be installed mostly at the wrist of manipulators, and are manufactured by several industries in the United States, Japan and Europe. Examples of use of these sensors, still quite limited in number if compared with their potentialities, are found in advanced manipulation and assembly.

Basically, the major part of these devices are transducers which measure forces/torques by means of the induced mechanical strains on flexible parts of their mechanical structure. The mechanical strains are in turns measured by elastomers (strain gauges), properly glued on the structure, that change their resistance according to local deformations, see a schematic drawing in Figure 2.1. Other technologies, like the optical one, have been used to realise force sensors [19], but have received less attention than the one based on elastomers.

For sensors based on the strain gauge technology, a linear mapping between the readings of the strain gauges, collected in a vector $\mathbf{s} \in \mathbb{R}^n$, and the applied wrench $\mathbf{w} = [\mathbf{f}^T, \mathbf{m}^T]^T \in \mathbb{R}^6$ is commonly assumed. This linear mapping is expressed by a matrix, called the *calibration matrix* $\mathbf{C} \in \mathbb{R}^{6 \times n}$:

$$\mathbf{w} = \mathbf{C} \mathbf{s}. \quad (2.1)$$

The basic capability of a force sensor is to measure the force/torque vector \mathbf{w} applied during the interaction with the object/environment. A major advance

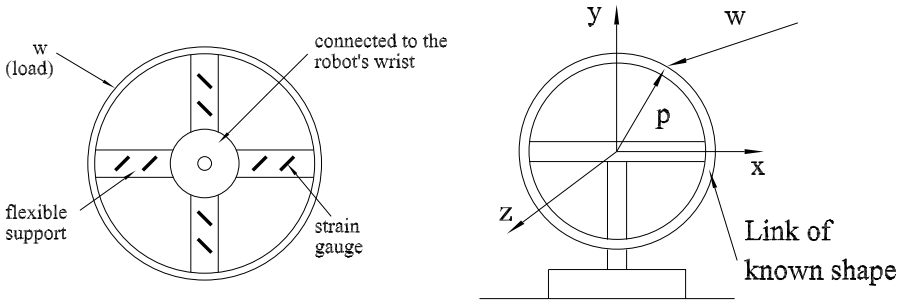


Figure 2.1. Conceptual scheme of a force/torque sensor (left) and a sensor connected to a link of known shape (right)

in force sensing has resulted after a paper of Salisbury in 1984 [33], where he proposed to use a force sensor for obtaining, besides the wrench \mathbf{w} , other significant information about the contact state. In particular, the position of the *contact centroid* [8] may be easily computed under some hypotheses. Later, Bicchi and Dario have extended this concept introducing the term *Intrinsic Tactile* (IT) sensing [7], [4].

The basic idea of the IT principle is illustrated in Figure 2.1. Given a force/torque sensor with known external shape and connected to a link of a manipulator, the basic feature of IT sensing is the possibility to determine, in case a contact is established between the link and an object, both the applied wrench \mathbf{w} and the position \mathbf{p} of the contact centroid on the surface of the link.

Successively, Eberman and Salisbury have extended the possibilities given by force sensors, considering not only “static” information about the wrench and the contact point location, but also the “dynamic” data obtained with a frequency analysis in different working conditions, [15]. A different idea concerning the dynamic use of force sensor has been more recently presented by Bicchi *et al* in [6]. Here, a force/torque sensor is used for measuring also impulsive forces and, since the sensor is *de facto* built with a flexible structure, the inversion problem is faced as an inversion dynamic problem typical of nonminimum phase systems.

One of the problems of the IT principle is the need of installing force sensors in the mechanical structure of the robot. While there are no relevant problems with “large” structures, the applicability of these sensors is quite limited for small devices, for example dexterous hands, due to technological difficulties of developing sensors of suitable size, both from the mechanical and the electronic point of view.

Only a few miniaturised 6-axis force sensors have been developed. Probably, the first was the one realised in 1985 at MIT [11], with a design based on

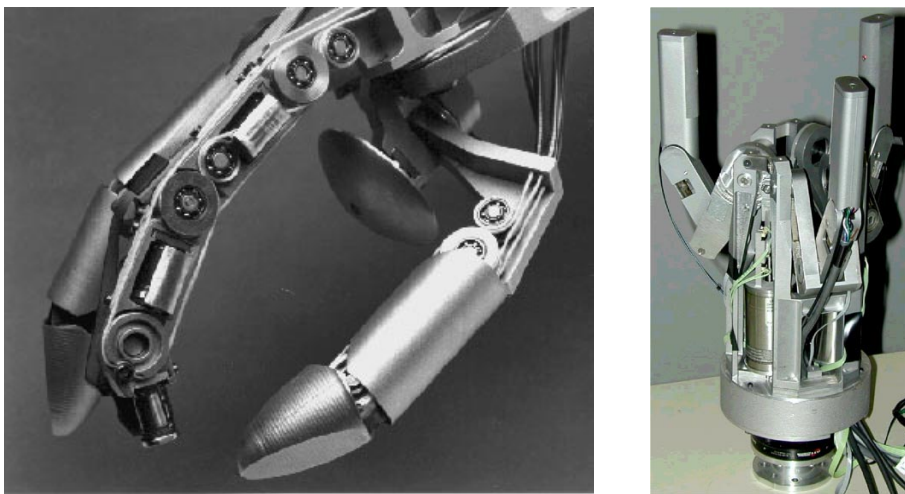


Figure 2.2. The U.B. Hand version II (left) and a three degrees of freedom gripper for space applications (right)

a Maltese cross and employing 16 strain gauges. This sensor has been used in the Salisbury Hand.

A second prototype has been realised in Pisa [7], with 6 strain gauges placed on a hollow cylinder. Note that 6 is the minimum number of strain gauges required to measure the 6 elements of the force/torque vector. In this sensor, the placement of the sensing elements has been optimised with respect to the condition number of the calibration matrix \mathbf{C} relating the strain gauges output and the applied forces [3]. A third miniaturised 6-axis force/torque sensor has been realised in the framework of the UB Hand project [9], see Figure 2.2. Due to very stringent mechanical constraints, this sensor has a very compact design, both in the mechanics and in the electronics, and employs 6 or 8 strain gauges glued on the sides of a square beam [12], see Figure 3.1. A description of this sensor is given with more details in Section 3.1. Another and more recent application of this type of sensor is in the framework of a 3-finger, 3-DOF gripper for space applications (Figure 2.2) where one sensor is present in each of the fingers to ensure robust grasp of unknown objects and control of the applied force, see [2]. After the initial developments of prototypes in research centers, nowadays some miniaturised force/torque sensors are also commercially available devices, see e.g. [1].

2.2 Tactile Sensors

Another very important class of sensing devices consists of tactile sensors. These are used for several purposes, such as shape recognition, contact point determination, pressure/force measurement. A number of tactile sensors have

been proposed in the literature, with several different solutions concerning the realisation features: optical, piezoresistive, piezoelectric, and so on. See the surveys given in [32], [31] and more recently in [26] for good overviews on technologies and applications.

Tactile sensors have been introduced in robotics since the late 70's. Nowadays, as the force sensors, also tactile sensors are commercially available devices. Probably, they represent the most commonly adopted sensorial class for industrial grippers, although often they are used as advanced 'on/off' devices to check whether a grasp or contact condition occurs.

Usually, they consist in a matrix (array) of sensing elements. Each sensing element is referred to as a *taxel* (from "tactile element"), and the whole set of information is called a *tactile image*. Main goal of this class of sensors is to measure the map of pressures over the sensing area.

The types of information that may be obtained from a tactile sensor are:

- *contact*: this is the most simple information given by the sensor, concerning the presence or absence of a contact;
- *force*: each sensing element provides an information related to the amount of locally applied force, which can be used in several manners for successive elaborations;
- *simple geometrical information*, i.e. position of the contact area, geometrical shape of the contact itself (planar, circular, and so on);
- *main geometrical features of the object*: by proper elaborations of the data of the taxels, it is possible to deduce the type of object in contact with the sensor, for example a sphere, a cylinder, etc. (data relative to the 3D shape);
- *mechanical properties*, such as friction coefficient, roughness, and so on: also thermal properties of the object may be measured by a tactile sensor;
- *slip condition*, i.e the relative movement between the object and the sensor.

2.3 Technology for Tactile Sensing

Since the introduction of tactile sensors in robotics in the 70's, several technologies have been adopted for their design, ranging from piezoresistive to magnetic and optical effects, [31, 26]. In the following, the main technological solutions are briefly described, illustrating some of the main advantages and disadvantages of each of them.

2.3.1 Resistive and conductive effect. Probably, this is the first and most popular method for transducing forces or pressures into electric signal, [36], [10]. Basically, this technique is based on the measurement of the electrical resistance of a material (elastomer) whose value depends on the locally applied force. Most common materials used as elastomers are made from carbon or silicon doped rubber. A grid of sensing points is built, where the resistance is measured, either across or through the elastomer, see Figure 2.3. These sensors are usually very robust, with respect to both durability

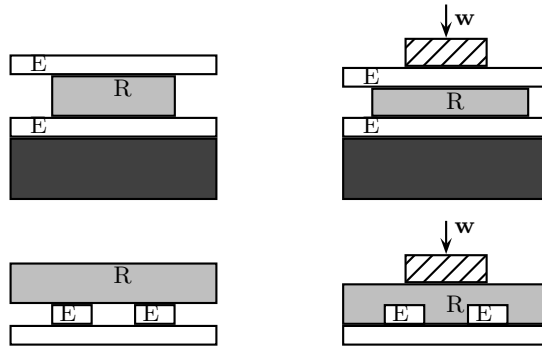


Figure 2.3. Tactile sensors based on resistive effect. R: rubber, E: electrodes

and mechanical effects, have a good dynamic range, and can be quite easily integrated with the acquisition electronics. On the other hand, the function relating the force and the resistance variation is not linear and may present some hysteresis effects. Moreover, a relevant number of wires is necessary (this is a problem common to all the matrix sensors).

2.3.2 Electromagnetic effect. Electromagnetic effects have been widely used in tactile sensing. Variation of a magnetic field, induced either by a mechanical movement of a part of the sensor or by an intrinsic property of a magneto-elastic material, can be easily detected and measured. Usually, these sensors have good performance, both in terms of linearity and hysteresis effects, may have the possibility of measuring the normal, tangential and torque component of the applied load, and are intrinsically robust. They are sensitive to extraneous electromagnetic fields.

2.3.3 Capacitive effect. The capacitance of a capacitor may change by varying the distance or more in general the coupling between its elements. This effect has been exploited in several manners for building tactile sensors, see for example Figure 2.4. Usually, these sensors have a linear response and

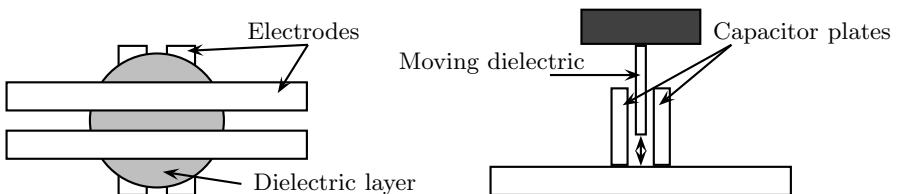


Figure 2.4. Tactile sensors based on capacitive effect

have an intrinsically robust design. On the other hand, there may be electrical disturbances, sensitivity to temperature, and dimensioning problems.

2.3.4 Piezoelectric effect. The piezoelectric effect of some materials, i.e. the generation of a voltage under the application of a pressure, can be easily exploited for tactile sensing. The generated voltage is proportionally related to the amount of pressure, and no external voltage is necessary. A similar effect, which has been used in some cases, is the pyroelectric effect, i.e. the generation of voltage with the change of temperature.

A possible drawback is that both these effects are dynamic, i.e. the voltage is generated during the application of a pressure or a change of temperature. In static condition, the effect is null. Another disadvantage is that often piezoelectric materials are basically crystals, such as quartz or ceramics: these materials are rather fragile and not suitable for this kind of applications. On the other hand, also sensors based on polymeric materials such as PVF₂ or PVDF have been successfully developed. Technological difficulties derive from the fact that each taxel need to be acquired by an individual electronics (i.e. the charge of each taxel must be acquired individually).

2.3.5 Optical effect. Tactile sensors based on optical effects exploit the modulation of light beams obtained by the deformation of transmitting material and/or reflection due to the object itself. Advantages of these sensors are the robustness with respect to external electrical disturbances or interference, the fact that the acquisition and elaboration electronics can be remotely located with respect to the sensor, the good compatibility with vision sensing technology. Drawbacks are mainly related to technological problems, such as the relevant number of fibers necessary for obtaining a good resolution.

2.3.6 Mechanical methods. A mechanical displacement originated by the contact is used as basic principle of some tactile sensors. Usually, these sensors are used as on/off devices, or as probe systems for measuring or welding apparatus. In order to measure the displacement, a number of different techniques have been employed, ranging from potentiometers to acoustic or electrical effects. This type of sensor usually is rather robust and reliable, but on the other hand gives some problems for building sensing arrays, and in any case for the relatively small achievable dimensions.

3 Examples of Force/Torque and Tactile Sensors

In the following, two prototypes of sensors for advanced robotic manipulations are briefly described. The first is a miniaturised force/torque sensor, suitable for installation in fingers of robotic hands, while the second is a tactile matrix sensor based on a conductive rubber. The sensors are described and examples of their use in advanced manipulation tasks are presented.

3.1 A Miniaturised Intrinsic Tactile Sensor

As already pointed out, the need of a force sensoriality is often a basic requirement for advanced robots. Moreover, in several applications the dimensional constraints of both the mechanical and the electronic part call for the development of sensors which can be easily installed and used in robots of small sizes.

Some of these devices have been realised at the University of Bologna, characterised by small dimensions and a good versatility. The first has been developed in the framework of the UB Hand project [9], where the need of a distributed force sensoriality led to the design and development of miniaturised 6-axis force sensors. More recently, this type of sensor has been also adopted in other prototypes of robotic grippers, such as 3-DOF grippers developed for space applications, [2].

The compactness of both the mechanical structure and the electronics, as well as the accuracy, resolution, and simplicity of the acquisition process have been main design and realisation objectives for these sensors [12]. The sensing element is realised by a light compliant mechanical structure with a parallelepiped shape, see Figure 3.1.a and Figure 3.1.b. In the first design, the sensor height, width and depth are respectively 21, 24, and 15 mm.

The base of the sensor is fixed to the link structure, while the other extremity is rigidly connected to the external shell. Contact wrenches produce a deformation of the sensing element proportional to the amount of force, and the deformation is measured by strain-gauges. As an example of installation, in Figure 3.1.c the arrangement of the sensors within the phalanges of one finger of the UB Hand is reported, see also Figure 2.2, while in Figure 3.1.d its application in the three-fingered gripper is shown.

In order to reduce at minimum the complexity of the sensor, the strain gauges are not used in a bridge configuration, and are serially connected. The compensation of the thermal effects on the resistance value is performed by the conditioning electronics, by means of a comparison of the voltage values given by the deformed strain gauges with the value of an undeformed one. A voltage reference is used as power supply.

These sensors have been already described in details, see e.g. [12], as well as their applications to robotic manipulation. In particular, with this sensor it is possible to: determine the position of the contact centroid on the finger surface; estimate the linear and rotational friction coefficient; control the applied forces/torques; verify whether the grasped object is close to a slipping condition. See [25], [33], [8], [12], [2], [30] for a description of these and other applications. Some results concerning the determination of the contact centroid are reported in Figure 3.2 with shells of different shapes.

In Figure 3.3 results for the measurement of the friction coefficient are shown: here the friction coefficient is computed as the ratio between the tangential and normal force components f_t, f_n :

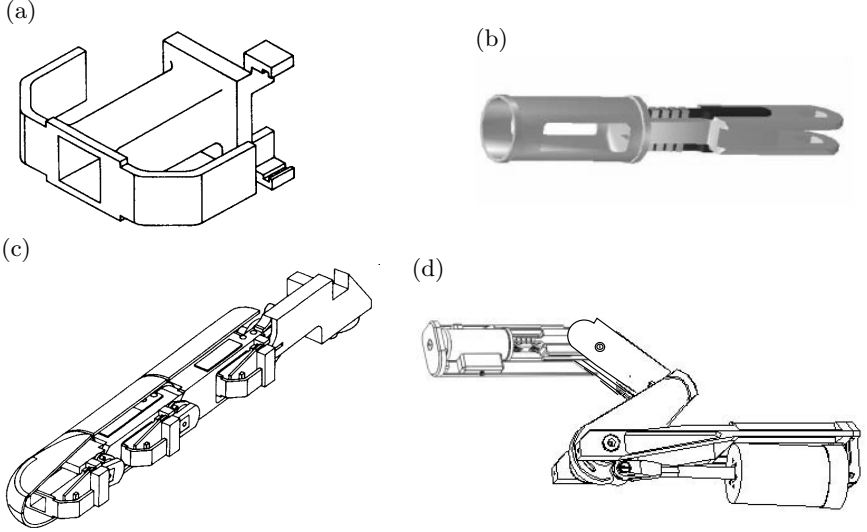


Figure 3.1. Two IT sensors (a) and (b), their placement within fingers of an anthropomorphic hand (c) and of a gripper (d)

$$\mu_s = \frac{f_t}{f_n}. \quad (3.1)$$

A similar approach can be used to determine the rotational friction coefficient, although in this case the coefficient is not independent on the size of the contact area, see [30] and [5] for more details.

Obviously, if the friction coefficient is known, this information can be used in order to apply proper control action if a slippage condition occurs.

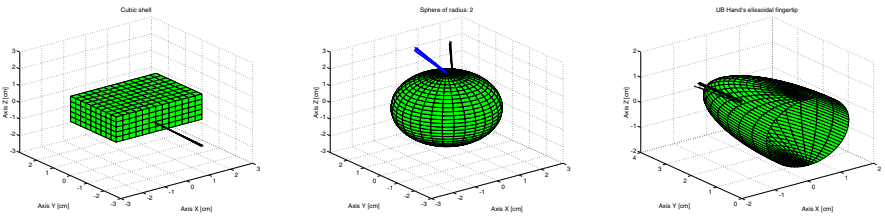


Figure 3.2. Measurement of the contact centroid and of the applied forces with links of different shape

3.2 A Tactile Matrix Sensor

An example of tactile sensing, used in the activities presented in the following, is the matrix tactile sensor developed at Delft University of Technology,

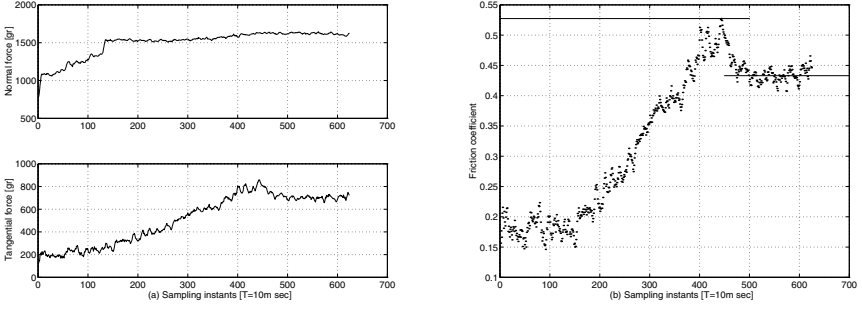


Figure 3.3. Determination of the linear friction coefficient: normal and tangential components (left) and their ratio (right)

[38]. As previously mentioned, a tactile matrix sensor measures the forces applied on it and how these forces are distributed over the sensor surface. Such a sensor in a robot system is able to provide information similar to that provided by human tactile sensors in the skin.

This tactile matrix sensor is based on conductive rubber. Note that since the adopted rubber is rather soft, it also enables a good grip on objects. The resistance of the rubber depends on the number of conductive particles inside the rubber (i.e. on the type of the rubber), on its geometry, and on the applied force.

The response of the rubber, as a result of the applied force, is highly non linear. Therefore, it is not possible to obtain an accurate force sensor with this technology, see Figure 3.4 for a typical characteristic of the rubber resistance as a function of the applied force. However, the rubber is capable of detecting the force distribution over the sensor area.

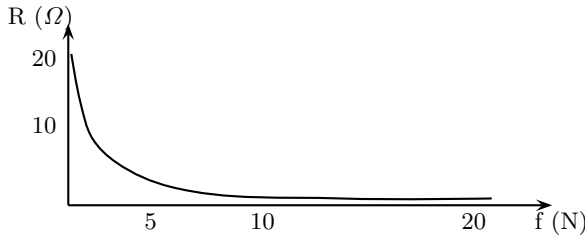


Figure 3.4. Resistance of the rubber as a function of the applied force f

In order to locally measure the resistance of the rubber, and therefore the value of the applied force, two electrodes are needed, since the resistance is measured by applying a current between the electrodes through the rubber. A pair of electrodes will be referenced to as a taxel.

The overall sensor consists in a matrix of 16×16 taxels, with a side length of about 2 cm, and a distance between two consecutive taxels of 1.25 mm. A printed circuit board has been used for the realisation of the electrodes, with a matrix (row/column) structure, see Figure 3.5.

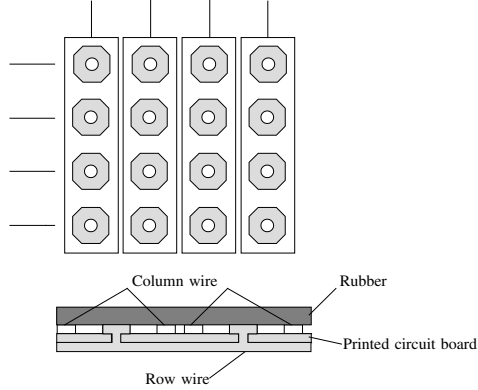


Figure 3.5. Tactile matrix sensor surface (4×4) with partial cross-section

The tactile sensor is able to detect 256 pressure levels and has a variable gain whereby the sensor sensitivity is automatically adjusted to ensure that maximum use is made of the available resolution. Some examples of images obtained by the tactile matrix sensor are given in Figure 3.6. Each grid point represents a taxel value between 0 and 255.

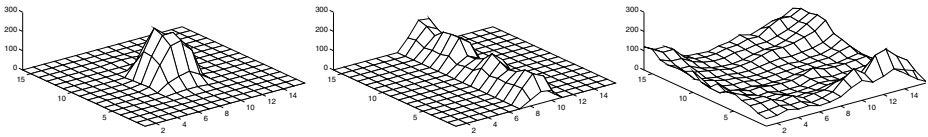


Figure 3.6. Some images of the tactile sensor

This sensor has been successfully used for typical applications, as object recognition. Different techniques have been implemented either based on neural network, see [14], [20], or on more conventional techniques based on successive data elaborations [27]. Moreover, more advanced applications have been developed, as also described in the following section.

4 Slip Detection

In the field of fine manipulation, consistent efforts have been devoted to designing sensors capable to detect a slip condition or even an incipient slip condition. The development of such sensors is very useful for example in the estimation of the friction coefficient of an unknown object during manipulation or also in exploiting the slip condition to improve manipulation. Accurate knowledge of the friction coefficient is particularly important for fine manipulation and surface tracking. In other words, e.g. when performing fine manipulation with fragile objects, it is important that the grasp force is maintained just above the minimum required to avoid the slippage of the object. During surface tracking, accurate and updated knowledge of friction conditions is important to avoid losing track of the object. Different methodologies have been adopted to achieve the above goal. In particular, one can distinguish two main approaches.

The first consists in calculating with a force sensor the friction coefficient by measuring the normal and the tangential forces when the object starts to slide. The main advantage of this approach is that it can be implemented without using dedicated sensors, although the measure is obtained in consequence of a relative movement of the object and the grasping device.

The second approach consists in the development of special sensors. These are mainly dedicated devices [13], [22], [37], designed and realised with the only goal of detecting a slip condition. Mostly, they are based on accelerometers placed in elastic materials which starts vibrating when slip begins. The main advantage of these sensors is that they provide a robust and a reliable measurement. On the other hand, they are dedicated devices and therefore increase the overall complexity of the sensorial system.

A different technique consists in exploiting the data provided by a tactile matrix sensor. Note that by using this type of sensor, capable of measuring also other significant features, such as the shape of the grasped object, contact points and a pressure-force distribution, the complexity of the overall sensorial system is limited.

4.1 Slip Detection with a Tactile Sensor

A method for slippage detection using a tactile matrix sensor and based on the rubber elasticity has been developed and experimentally tested, see [28], [21]. As a matter of fact, before the object starts to slip the rubber is stretched: this deformation can be interpreted as a signal of an incipient slip. In fact, a slip can be detected by implementing a frequency analysis of the center of the force distribution and by testing the low frequency components.

This approach employs the elasticity of the rubber. In particular, when a tangential force is applied to the object, the local shape of the rubber is modified. One consequence is that the center of distribution changes its position. By implementing a Fourier analysis and by testing the low frequencies,

it is possible to recognise the deformation that precedes the real slip. It is important to note that this effect is independent on the size and type of the object.

In detail, the algorithm involves the following steps. A new position of the center of distribution is calculated every sampling period as:

$$x_c = \frac{\sum_{i=1}^n \phi_i x_i}{\sum_{i=1}^n \phi_i} \quad y_c = \frac{\sum_{i=1}^n \phi_i y_i}{\sum_{i=1}^n \phi_i} \quad n = 256$$

where ϕ_i is the pressure value of the i -th taxel (proportional to the local rubber resistance), and x_i, y_i are its coordinates on the tactile matrix. The Cartesian coordinates of the center of distribution are then transformed in polar coordinates (ρ, θ) and, for the sake of simplicity, only the component $\rho = \sqrt{x_c^2 + y_c^2}$ is considered. Then, a FFT of ρ is computed as:

$$R(k) = \sum_{i=0}^{N-1} \rho_i e^{j2\pi i k / N} \quad k = 1, 2, \dots, N.$$

When no slip is present, and disregarding the component at zero frequency, the spectrum does not give any significant information since it is dominated by noise, see Figure 4.1.a.

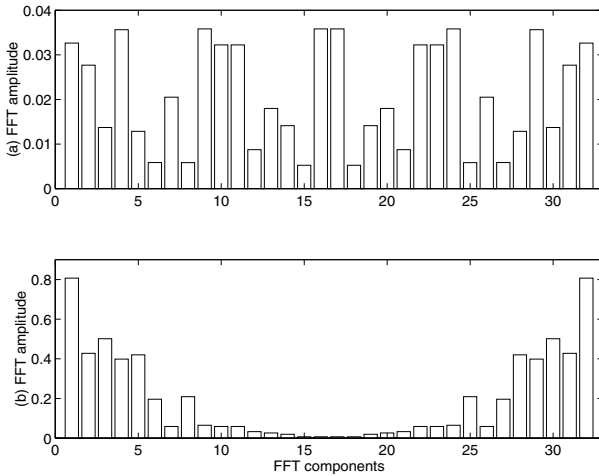


Figure 4.1. Typical spectrum without (a) and with (b) slip

On the other hand, when a tangential force is applied and the rubber initiates to deform, just before the object starts to slip the low frequency components dominate the spectrum, which assumes a well definite shape, see Figure 4.1.b. This effect is due to the shift of the center of distribution at low speed. It is important to note that this effect is present *before* the

object actually starts to slip (i.e. when the object is still stuck to the rubber) and also if the object completely covers the sensor. Once the object starts to slip, the center of gravity is stable again and the DC component dominates, generating a spectrum as in Figure 4.1.a. This means that this technique is capable only to measure the initial phase of slip since it gives, in practice, the same results when the object is fixed and when it moves with a constant velocity due to the application of a constant force.

In Figure 4.2 results of a typical experiment, obtained with the setup described in Section 5, are reported. In particular, in Figure 4.2.a the movement of a grasped object is shown, as measured with a precise position sensor, along with the first 3 components of the FFT. It may be easily seen that the information given by the spectrum allows detecting the incipient slip of the object. Similarly, also a pure rotational slippage of a grasped object may be detected, see Figure 4.2.b, although in this case the detection of the rotation is most likely due to nonlinearities of the rubber more than by a motion of the center of distributions.

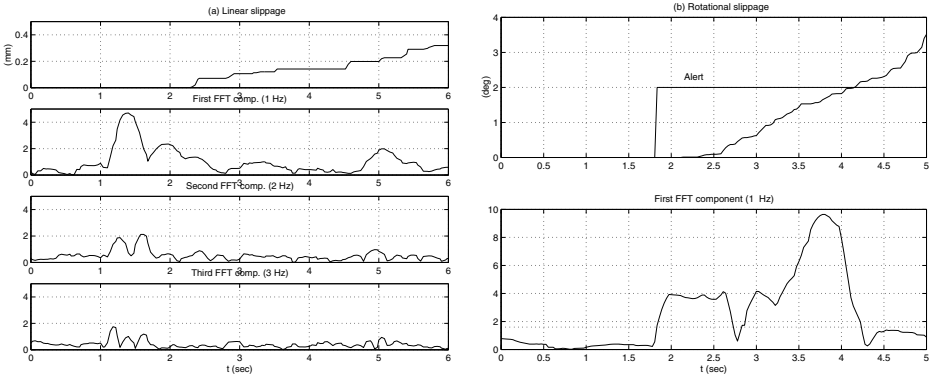


Figure 4.2. Linear (a) and rotational (b) slip detection

The experiments demonstrate that this technique is very efficient in detecting the initial phase of the slip, while on the other hand gives poor results during the motion of the object.

4.2 Slip Detection with a Tactile and Force/Torque Sensor

By using a tactile sensor alone, as discussed in the previous section, it is possible to detect the linear slip between an object and the grasping manipulator. In a more general context, it may be of interest to detect also the rotational slip which can occur during manipulation. In this respect, a force/torque or a tactile sensor alone are not adequate, since on one hand the ‘rotational’ friction coefficient in general is not expressed by a simple linear relationship

as (3.1) and on the other the knowledge of the applied wrench is of interest. In the following, an activity for integrating the two sensors is summarised. For more details refer to [30], [5], [29].

In case of rotation, in general a nonlinear relationship exists between the applied normal force f_n and the torque m_f [24]. This relationships may be expressed as

$$m_f = 0.59\mu_s\alpha^{1/3}f_n^{4/3} = \beta'_sf_n^{4/3} \quad (4.1)$$

where α is a parameter depending on the geometry and on the elastic properties of the contacting bodies. Equation (4.1) may be approximated, in suitable ranges of torques m_f , by

$$m_f = \beta_sf_n. \quad (4.2)$$

The friction coefficient β_s (or β'_s) depends on the size of the contact area, and this geometrical information cannot be provided by the force/torque sensor, although it can provide, for a given contact, information about the ratio m_f/f_n .

In the literature, different models have been defined for describing the effects of friction with linear and torsional loads in the contact area, see e.g. [16], [23], [18]. For practical applications, important issues are the simplicity and the adequacy of these models, that must be evaluated in real time. Among others, the models illustrated in [23] represent quite simple tests for real-time implementation. In particular, the linear model

$$\frac{f_t}{\mu_s} + \frac{\tau}{\beta_s} = f_n \quad (4.3)$$

and the elliptical one

$$\frac{f_t^2}{\mu_s^2} + \frac{\tau^2}{\beta_s^2} = f_n^2 \quad (4.4)$$

are both simple and sufficiently adequate for describing the phenomena of interest, and give very simple conditions for checking a slip condition. On the other hand, implicit drawbacks of these models are the fact that they are based on global information (i.e. the overall force and the torque) and that it is not simple to have a good estimation of the rotational friction coefficient β_s , since it depends on the size of the contact area (non measurable with force sensors alone).

4.3 Rotational and Translational Slip Detection

When only linear forces are applied at the contact, if a Coulomb friction model is considered with static coefficient μ_s , from (3.1) a simple condition to check whether a relative motion between two bodies takes place consists in verifying the inequality

$$f_f/f_n \leq \mu_s \implies \left(\sqrt{f_{fx}^2 + f_{fy}^2} \right) / f_n \leq \mu_s. \quad (4.5)$$

This simple method has been successfully used in several experiments for slip detection and control using a IT sensor, see e.g. [25] and [4].

On the other hand, this condition is not sufficient in case the forces acting on the object generate a torque component on it. As already pointed out, in this case a more general relationship must be defined, taking into account also the geometry of the contact area. For these reasons, we consider in the following the integrated use of a IT and a tactile matrix sensor, see also [30], [5], [29].

In general, the motion of a rigid body (in contact with a surface) is a rotation about a point called *Center of Rotation* (COR). A particular case is when this point is at an infinite distance, and the motion is a pure translation. When the body starts to slide, the directions of all the tangential forces ϕ_f on the contact surface cause the body to rotate about the COR. This means that at the limit condition these friction force components (applied at the generic point \mathbf{p}_i) have direction perpendicular to the line connecting \mathbf{p}_i to the COR, i.e. $\phi_f^T(\mathbf{p}_i - \mathbf{p}_c) = 0$, see Figure 4.3.

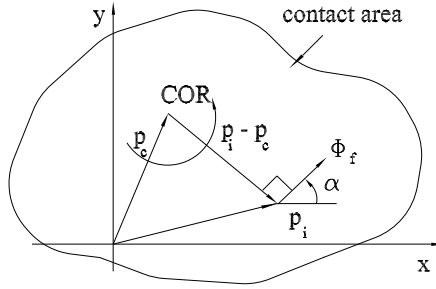


Figure 4.3. The direction of the local friction force ϕ_f at the limit condition is perpendicular to $\mathbf{p}_i - \mathbf{p}_c$

A practical approach to test the slip condition is reported in the following. Let us suppose that the COR position $\mathbf{p}_c = [x_c \ y_c]^T$ is known (with respect to a given reference frame \mathcal{F}). Moreover, we consider to use both a force/torque sensor (by means of which we obtain the normal and tangential force f_n and f_t , and the overall torque m_t) and tactile matrix sensor (by means of which we measure the size and shape of the contact area A and the distribution of local pressure). From the combination of the two sensors, it is possible to evaluate also the local normal forces ϕ_n in each taxel of the tactile sensor.

Considering the ‘discrete’ nature of the tactile sensor, and assuming to be at the limit condition, it is possible to compute the torque m_c (about z) given by the friction forces acting on A with respect to the COR as:

$$\begin{aligned}
m_c &= \sum_{i=1}^{256} |\phi_{f,i} \times (\mathbf{p}_i - \mathbf{p}_c)| \\
&= \sum_{i=1}^{256} \mu_s \phi_{n,i} \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2}.
\end{aligned} \tag{4.6}$$

Then, it is possible to compare m_c with the torque m_t measured by the force sensor. No (rotational) motion takes place between the two bodies if

$$m_t \leq m_c = \sum_{i=1}^{256} \mu_s \phi_{n,i} \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2}.$$

Since also an external force f_t can be applied to the object, by defining $n_t = |m_t + f_t \times \mathbf{p}_c|$, the condition to check whether slippage takes place becomes

$$n_t \leq m_c = \mu_s \sum_{i=1}^{256} \phi_{n,i} \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2}. \tag{4.7}$$

From a theoretical point of view, in this manner it is possible to verify whether a slippage takes place, verifying at the same time (with the COR position) if it consists of a rotation and a translation or of a pure translation.

From the above, since the quantities m_t , f_t , $\phi_{n,i}$ are known from the two sensors, it follows that the problem reduces to the determination of the COR position \mathbf{p}_c , from which, by means of (4.6), it is possible to compute m_c .

In practice, once the external tangential force $\mathbf{f}_t = [f_{tx} \ f_{ty}]^T$ acting on the body is known, as well as the distribution of the local force components ϕ_f and ϕ_n and the geometry of the contact area A , the COR position \mathbf{p}_c can be computed from the following equations, see also Figure 4.3,

$$\begin{cases} f_{tx} &= - \sum_{i=1}^{256} \frac{\mu_s \phi_{n,i} (y_i - y_c)}{\sqrt{(x_i - x_c)^2 + (y_i - y_c)^2}} \\ f_{ty} &= \sum_{i=1}^{256} \frac{\mu_s \phi_{n,i} (x_i - x_c)}{\sqrt{(x_i - x_c)^2 + (y_i - y_c)^2}} \end{cases} \tag{4.8}$$

being x_i, y_i the position of the i -th taxel. Equations (4.8) allows computing the unknown COR position, i.e. x_c, y_c . Note that we have assumed in (4.8) a constant friction coefficient in each point of the contact surface. Moreover, note that these equations are valid only at the limit condition for slippage, i.e. when $|\phi_f| = \mu_s \phi_n$. On the other hand, the computation of the COR position makes sense only after the slip motion begins, being null the relative motion of the objects beforehand. Finally, the possibility that the COR coincides with one of the taxels of the tactile sensor must be explicitly considered, since in this case (4.8) cannot be used.

In order to solve the nonlinear equations in (4.8) in the unknowns x_c, y_c , a recursive technique has been adopted, similar to methods for the solution

of the inverse kinematic problem of robot manipulators, [35].

The relationship in (4.7) represents an extension of (4.5) to the case of both forces and torques applied at the contact, and is more general than (4.3)–(4.4) since it takes into consideration the size of the contact area and avoids the computation of the rotational friction coefficient β_s .

5 Experimental Results

The techniques described in the previous sections have been experimentally verified by using laboratory setups integrating the force/torques and the tactile sensors. In particular, the first setup consists in a planar 2-DOF ‘finger’, with a ‘fingertip’ equipped with the IT and the tactile sensors, a fixed surface for holding an object, and an external motor, with a precision position sensor, for applying forces/torques to the object and measuring its displacement, while the second is a two 1-DOF fingers gripper, again equipped with the two sensors, see Figure 5.1.

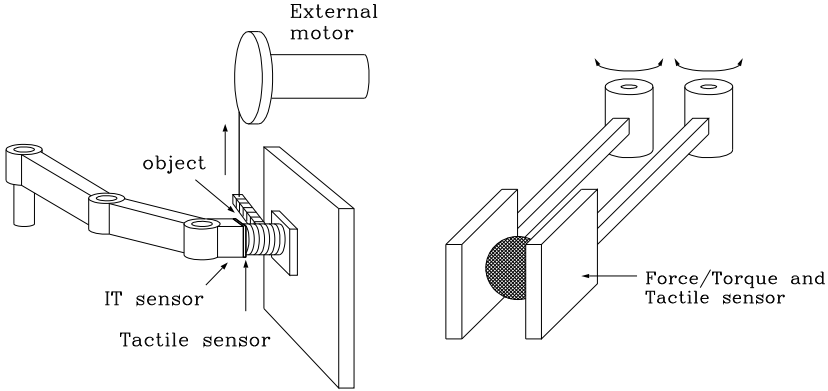


Figure 5.1. The two laboratory experimental setups

The results reported in the following refer to the detection and control of linear slippage, based on the IT sensor alone, and to the rotational/translational slip detection, involving the use of the IT sensor alone and the integration of the IT and the tactile matrix sensor.

5.1 Translational Slip Detection and Control

If the friction coefficient is known, e.g. by means of a measurement as in Figure 3.3, it is possible to use it in manipulation strategies in order, for example, to prevent slippage of the object.

The simplest control strategy for avoiding slippage is to use (4.5) to compute in real time the normal force f_n needed to balance the applied tangential force f_t . A more convenient solution consists in adopting a conservative scaling factor $\sigma \leq 1$ for the friction coefficient. Moreover, it may be necessary to bound the computed value of f_n in order to neglect measurement noise and to avoid the application of high forces. Therefore, the normal force f_n is computed as

$$\begin{cases} f_{n1} = \frac{f_t}{\sigma\mu_s}, \\ f_{n2} = \max\{f_{n1}, f_{nom}\} \end{cases} \quad f_n = \min\{f_{n2}, f_{max}\}$$

where f_{nom} is the nominal force to be applied when no tangential components are present, and f_{max} is the maximum applicable value. Results obtained with

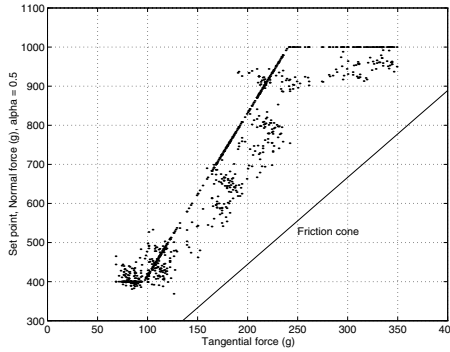


Figure 5.2. Linear slip avoidance with a conservative values of μ_s ($\sigma = 0.5$)

this technique are shown in Figure 5.2. In the figure, the value $\sigma = 0.5$ has been used. Each dot represents the applied normal force as a function of the tangential force, the diagonal solid line represents the border of the friction cone: the slip is always avoided. In the experiment, the measured friction coefficient was $\mu_s = 0.45$. A similar strategy to prevent slippage has been proposed in [4].

5.2 Rotational and Translational Slip Detection and Control

Although even the rotational slippage may be detected by using a force/torque sensor alone, this possibility is intrinsically limited by the fact that the dimension of the contact should be known in advance, and moreover it should

be constant during manipulation. In the following, an integrated use of a tactile and a force/torque sensor is described, illustrating some laboratory experiments of the technique discussed in Section 4.3.

5.2.1 Measurement of the COR position. In general the COR position is not known and must be computed in real-time. Results obtained in this case are reported in Figure 5.3, reporting the displacement (note that this measure is a composition of both a rotational and a linear motion), the ‘slip’ signal, and the torques m_f (computed by the algorithm) and m_t (measured by the IT sensor). Figure 5.4 shows the computed COR position on the tactile pad.

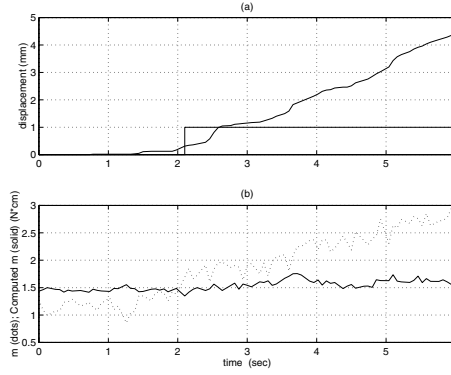


Figure 5.3. Experiment with unknown COR position: (a) rotational displacement and result of condition (4.7) (1 = slippage); (b) torques m_f (solid) and m_t (dotted)

5.2.2 Rotational and translational slip control. Since it is possible to detect the slip between the two contacting bodies, it is possible to apply a control action similar to the one presented for the translational case. The condition to be checked is (4.7). For the sake of simplicity, we assume that the variation δf_n of the normal force f_n applied by the robotic finger is uniformly distributed on the contact area A , i.e. all the components $\phi_{n,i}$ change of the same amount $\delta\phi_n = \delta f_n / N$. Moreover, we define $\delta f_n = \sum_{i=1}^{256} \delta\phi_n$. Therefore, from (4.7) we obtain

$$n_t \leq \sum_{i=1}^{256} \mu_s \delta\phi_n \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2}.$$

With the introduction of a conservative factor σ , in case (4.7) does not hold we compute the desired variation δf_n of the normal force as

$$\delta f_n = \frac{n_t - m_c}{\sigma \mu_s \sum_{i=1}^{256} \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2}} \quad (5.1)$$

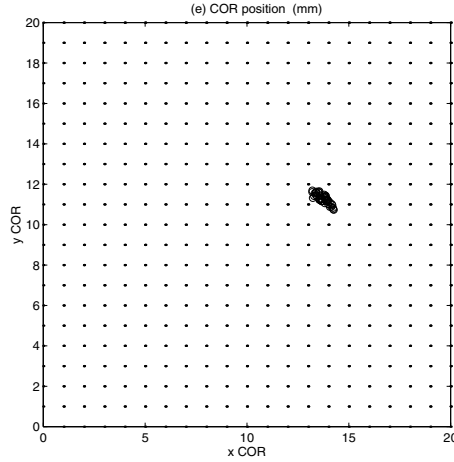


Figure 5.4. COR position on the tactile pad

Results obtained with the above control strategy are shown in Figure 5.5, from which it may be appreciated the increase of the grasping force (the normal force f_n) when the external disturbance is applied, and the fact that slippage is stopped.

Note that the fact of using both the sensors allows avoiding the rotational friction coefficient measurement, which depends on the size of the contact area A and therefore may change during task execution. Moreover, note that the object may be constrained using smaller values of the normal force (the technique based only on the IT sensor provides more conservative control actions).

5.3 Manipulation Exploiting Friction

The aim of the results discussed in this section represents a rather different approach to manipulation. The basic observation is that even in normal tasks the human beings normally exploit the friction in order to manipulate an object. Moreover, it can be observed that the capability of manipulating an object by taking advantage of friction may actually increase the dexterity of a given robotic device. Therefore, simple devices could be used to execute, although within certain limits, dexterous manipulations. For example, imagine a standard robotic gripper with only one degree of freedom: a grasped object could be ‘moved’ once it is grasped by exploiting the friction effect (and the external gravity force) by imposing a controlled slip.

On the basis of the results illustrated in the previous sections for the detection of a rotational slippage, based on the comparison of the disturbance and the friction torque, the feasibility of executing a controlled slip of a

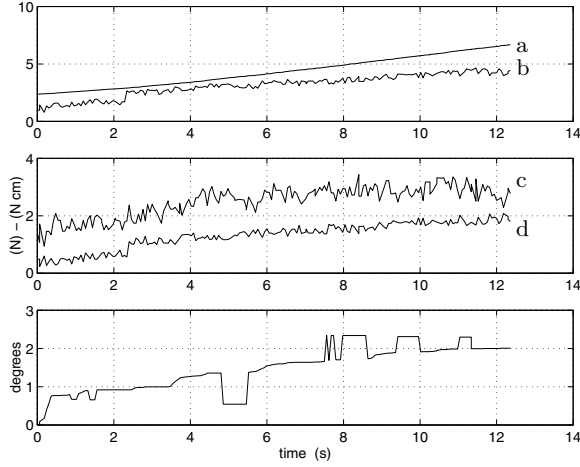


Figure 5.5. Rotational and translational slip control with both sensors: set point (a) and normal force (b), disturbance force (c) and torque (d); measured angle of displacement

grasped object has been tested with the setup of Figure 5.1. The goal was to let a cylindrical object rotate of a given amount by properly decreasing the grasping force.

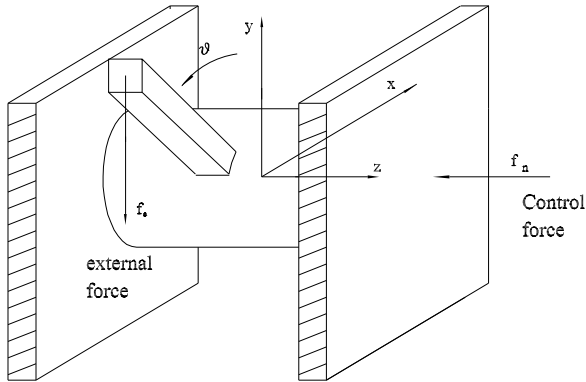


Figure 5.6. Controlled rotation with friction

A schematic drawing illustrating the experiment is reported in Figure 5.6. The object is grasped with a normal force f_n (controlled variable), and an external constant force f_a is applied. The value of the normal force is decreased until the comparison between the friction force and the external torque generated by force f_a indicates that a slip condition is generated. At this point,

a computation of the rotational movement of the object is performed and the normal force is changed accordingly. When the desired angular position is reached, the value of f_n is increased in order to firmly hold the object. This logic is summarised in Figure 5.7. Results of this approach are shown in Figure 5.8 and Figure 5.9.

Note that in this simple case from the geometry of the grasp, assumed known, it is possible to write

$$\theta = \arccos \frac{m_z}{f_a l} \quad (5.2)$$

where the force f_a is constant (in the reported experiment $f_a = 0.86$ N). It is therefore possible to determine the angular position θ on the basis of a measure of the IT sensor.

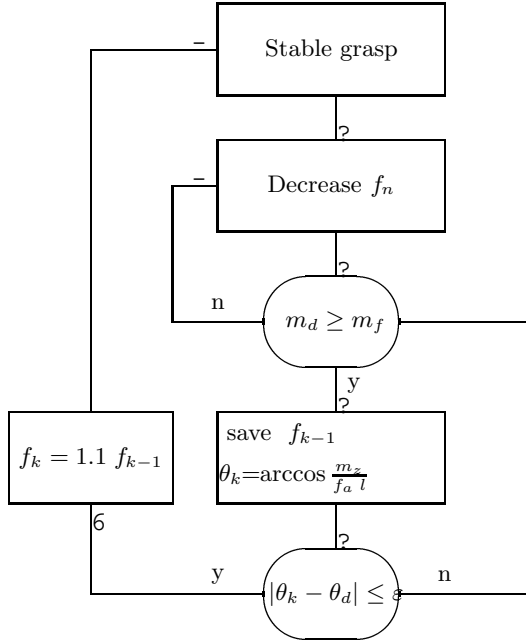


Figure 5.7. A slip control strategy

The control strategy is summarised as follows, see also Figure 5.7:

- the normal force f_n is decreased and a slip condition is detected by comparison of the external and friction torque;
- when the slip condition is detected, the current values of f_n and of θ , from (5.2), are saved;

- the value of θ is compared with the desired set-point; if the error is less than a threshold, the normal force is increased, otherwise it is decreased.

The results shown in Figure 5.8 and Figure 5.9 have been obtained by imposing the initial normal force $f_n = 4N$. The desired rotation is $\theta_d = 45$ deg, and the value for the threshold in Figure 5.7 is $\varepsilon = 0.5$ deg.

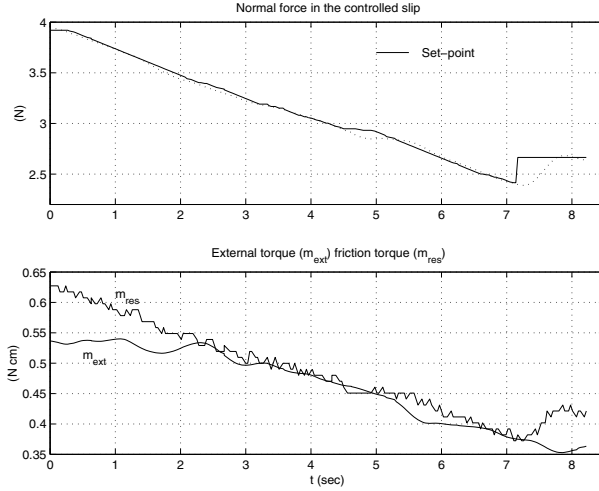


Figure 5.8. Normal force and torques

In Figure 5.8 the external forces are shown. Initially, the normal force is decreased until the slip begins. At this point, the algorithm summarised in Figure 5.7 is applied. When the desired angular position is reached, the normal force is increased in order to stop the slippage.

6 Conclusions

This chapter has described the force and tactile sensing technologies and some of their applications in advanced robotic manipulation. Force and tactile sensors developed at the Universities of Bologna and Delft have been considered as significant examples, and have been used on two robotic setups for illustrating their effectiveness. The presented results refer to different aspects of manipulation, from the estimation of the friction coefficient, to slip detection, to grasp control strategies aimed to minimising the grasp forces or optimising the hand/finger posture. Other uses of these sensors, reported only marginally, concern the implementation of force control strategies for dexterous manipulation. Moreover, research activities devoted to the integration of the two devices in a single sensor and to further exploitation of the provided information have been presented.

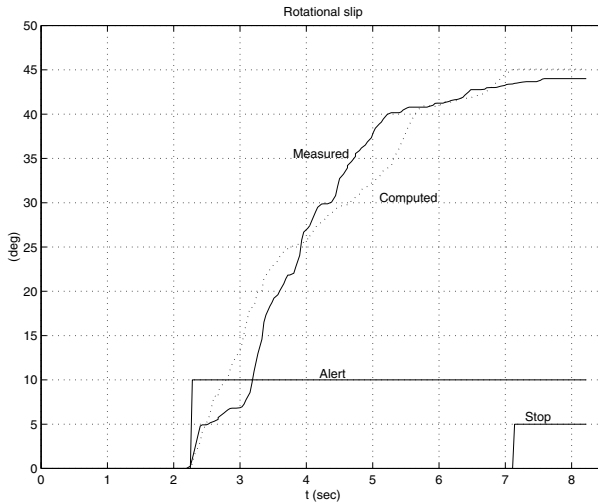


Figure 5.9. Rotational motion of the grasped object

From the results discussed here, representing only some aspects of the potentialities and use of these type of sensors, and that obviously only reflect the author's point of view about these technologies, it clearly emerges that this field is still in a growing stage, and that further research is both necessary and of relevant interest for the robotic community, especially considering the current use of robot technologies in a number of different applications in non structured environments.

Acknowledgements

The activity described here is based on the work of several researchers at DEIS and DIEM of University of Bologna. In particular, the author thanks the contributions of C. Bonivento, L. Marconi, A. Tonielli, G. Vassura, A. Eusebi, S. Andreotti: they are gratefully acknowledged for their unvaluable efforts. The author is also grateful to W. Jongkind and E. Holweg (Delft University of Technology) for the cooperation in part of these activities.

References

1. ATI, <http://www.ati-ia.com>
2. Biagiotti L, Melchiorri C, Vassura G 2000 Experimental activity on grasping objects in free-floating conditions. In: *6th ESA Workshop on Advanced Space Technologies for Robotics and Automation* Noordwijk, NL, 3.3-9

3. Bicchi A 1989 *Tools and Methods for Robot Hand Control* PhD Thesis (in Italian), University of Bologna
4. Bicchi A 1990 Intrinsic contact sensing for soft fingers. In: *Proceedings of 1990 IEEE International Conference on Robotics and Automation* Cincinnati, OH, pp 968–973
5. Bicchi A, Bergamasco M, Dario P, Fiorillo A 1988 Integrated tactile sensing for gripper fingers. In: *Proceedings of 7th International Conference on Robot Vision and Sensory Control* Zurich, CH, pp 340–349
6. Bicchi A, Caiti A, Prattichizzo D 1999 Optimal design of dynamic multi-axis force/torque sensor. In: *Proceedings of 38th IEEE Conference on Decision and Control* Phoenix, AZ, pp 2981–2986
7. Bicchi A, Dario P 1987 Intrinsic tactile Sensing for Artificial Hands. In: Bolles R, Roth B (Eds) *Robotics Research* MIT Press, Cambridge, MA, pp 83–90
8. Bicchi A, Salisbury J K, Brock D L 1990 *Contact Sensing from Force Measurements* AI Memo No 1262, MIT, Cambridge, MA
9. Bonivento C, Melchiorri C 1993 Towards dexterous manipulation with the U.B. hand II. In: *Preprints of 12th IFAC World Congress* Sydney, AUS, pp X:185–188
10. Briot M 1979 The utilization of an artificial skin sensor for the identification of solid objects. In: *Proceedings of 9th International Symposium on Industrial Robots* Washington, DC, pp 529–547
11. Brock D L, Chiu S 1985 Environment perception of an articulated robot hand using contact sensors. In: *Proceedings of ASME Winter Annual Meeting* Miami, FL, pp 228–235
12. Cicchetti A, Eusebi A, Melchiorri C, Vassura G 1995 An intrinsic tactile force sensor for Robotic manipulation. In: *Proceedings of 7th International Conference on Advanced Robotics* Sant Feliu de Guixols, S, pp 889–894
13. Cuttino J F, Huey C O, Taylor T D 1988 Tactile sensing of incipient slip. In: *Proceedings of USA–Japan Symposium on Flexible Automation* Minneapolis, MN, pp 547–555
14. den Braven S 1994 *Shape Recognition Using a Tactile Matrix Sensor* Internal Report (in Dutch), Delft University of Technology
15. Eberman B, Salisbury J K 1994 Application of change detection to dynamic contact sensing. *International Journal of Robotics Research* 13:369–394
16. Erdmann M 1994 On a representation of friction in configuration space. *International Journal of Robotics Research* 13:241–271
17. Gorinevsky D M, Formalsky A M, Schneider A Yu 1997 *Force Control of Robotic Systems* CRC Press, Boca Raton, FL
18. Goyal S 1989 *Planar Sliding of a Rigid Body with Dry Friction: Limit Surfaces and Dynamics of Motion* PhD Thesis, Cornell University
19. Hirzinger G, Brunner B, Dietrich D, Heindl J 1993 Sensor-based robotics — ROTEX and its telerobotic features. *IEEE Transactions on Robotics and Automation* 9:649–663
20. Holweg E G M 1994 Object recognition using a tactile matrix sensor. In: *Proceedings of European Robotics and Intelligent Systems Conference* Malaga, S, pp 1379–1383
21. Holweg E G M, Hoeve H, Jongkind W, Marconi L, Melchiorri C, Bonivento C 1996 Slip detection by tactile sensors: Algorithms and experimental results. In: *Proceedings of 1996 IEEE International Conference on Robotics and Automation* Minneapolis, MN, pp 3234–3239
22. Howe R, Cutkosky M 1989 Sensing skin acceleration of slip and texture perception. In: *Proceedings of 1989 IEEE International Conference on Robotics and Automation* Scottsdale, AZ, 1989, 145–150

23. Howe R D, Kao I, Cutkosky M R 1988 The sliding of robot fingers under combined torsion and shear loading. In: *Proceedings of 1988 IEEE International Conference on Robotics and Automation* Philadelphia, PA, pp 103–105
24. Jameson J W 1985 *Analythic Techniques for Automated Grasp* PhD Thesis, Stanford University
25. Jongkind W, Melchiorri C 1996 Tactile sensing for robotic manipulation. In: Tarn T J, Xi N (Orgs) *Workshop on Force and Contact Control in Robotic Operations: Theory and Applications, 1996 IEEE International Conference on Robotics and Automation* Minneapolis, MN, pp 8–32
26. Lee M H, Nicholls H R 1999 Tactile sensing for mechatronics – A state of the art survey. *Mechatronics* 9:1–31
27. Marconi L 1995 Tactile sensor data elaboration for object recognition and interpretation. In: *2nd TELEMAT Students Congress* Noordwijkerhout, NL
28. Marconi L, Melchiorri C 1996 Incipient slip detection and control using a rubber-based tactile sensor. In: *Preprints of 13th IFAC World Congress* San Francisco, CA, pp 475–480
29. Melchiorri C 1999 Translational and rotational slip detection and control in robotic manipulation. In: *Preprints of 14th IFAC World Congress* Beijing, PRC, pp B:161–166
30. Melchiorri C 2000 Slip detection and control using tactile and force sensors. *IEEE Transactions on Mechatronics* 5:235–243
31. Nicholls H R, Lee M H 1989 A survey of robot tactile sensing technology. *International Journal of Robotics Research* 8(3):3–30
32. Pugh A 1986 *Robot Sensors, Vol 2: Tactile and Non-Vision* Springer, London, UK
33. Salisbury J K 1984 Interpretation of contact geometries from force measurements. In: Brady M, Paul R (Eds) *Robotics Research* MIT Press, Cambridge, MA, pp 565–577
34. Salisbury J K 1987 Whole-arm manipulation. In: Bolles R, Roth B (Eds) *Robotics Research: The 4th International Symposium* MIT Press, Cambridge, MA, pp 183–189
35. Sciavicco L, Siciliano B 1988 A solution algorithm to the inverse kinematic problem for redundant manipulators. *IEEE Journal of Robotics and Automation* 4:403–410
36. Snyder W E, Clair Jr S 1978 Conductive elastomers as a sensor for industrial parts handling equipment. *Transactions on Instrumentation and Measurement* 27(1):94–99
37. Son J S, Monteverde E A, Howe R D 1994 A tactile sensor for localizing transient events in manipulation. In: *Proceedings of 1994 IEEE International Conference on Robotics and Automation* San Diego, CA, pp 471–476
38. Svihus J 1993 *Tactiele Matrix Rubber Sensors* MS Thesis (in Dutch), Delft University of Technology
39. Taddeucci D, Laschi C, Lazzarini R, Magni R, Dario P, Starita A 1997 An approach to integrated tactile perception. In: *Proceedings of 1997 IEEE International Conference on Robotics and Automation* Albuquerque, NM, pp 3100–3105
40. Waldron K J 1987 Force and motion management in legged locomotion. *IEEE Journal of Robotics and Automation* 3:214–220

On Control of Flexible Robots

Antonio Ficola, Mario Luca Fravolini, and Michele La Cava

Dipartimento di Ingegneria Elettronica e dell'Informazione
Università degli Studi di Perugia
<http://www.diei.unipg.it/RICERCA/rautom.htm>

This chapter deals with control of robots with flexible links. The most recent results about modelling and control of flexible robots are examined; in particular, the results obtained by the authors and concerning predictive control are described in detail.

1 Introduction

A mechanical structure with holonomic constraints can be represented by the following equations:

$$B(q)\ddot{q} + h(q, \dot{q}) + Kq + g(q) = \begin{bmatrix} \tau \\ 0 \end{bmatrix} \quad q \in R^n, \tau \in R^m \quad (1.1)$$

where $B(q)$ is a symmetric positive definite matrix, $h(q, \dot{q})$ includes the Coriolis and centrifugal generalised forces, K is a constant positive (semi)definite matrix that model the elasticity, $g(q)$ is the vector of gravitational generalised forces, τ is the control vector.

In case of robots with elastic joints and/or flexible links the number of control inputs is smaller than the number of Lagrangian coordinates $m < n$, making the system underactuated. In fact, the elements of the flexible links are coupled by the elastic forces/torques Kq , which can be considered generated by proportional regulators with zero set points. Therefore, a flexible link without gravity has a unique equilibrium configuration of the neutral axis, determined only by the term Kq ; in this case the elasticity reduces the degree of underactuation. The term $g(q)$ can be regarded as a further input or a disturbance.

In fact, the “underactuation” of the flexible robots is not of the most general kind. For instance, in case of rigid walking machines, only the gravity can be used as a further input; in case of rigid robots moving in a horizontal plane with some underactuated joints, not even the gravity is present, causing a “true underactuation”. Therefore, the wider class of underactuated holonomic systems comprises systems without gravity and elastic elements, and includes the class of flexible robots.

2 Modelling

Modelling structures with flexible elements requires a preliminary examination of a unique flexible element, since a distributed parameters body is more complex to be described than a rigid body. For this reason, in the following section some results in the analysis of a single flexible link are presented.

2.1 Single Link

The models are usually developed referring to the Saint Venant assumptions: i) the material behaves according the Hooke's law; ii) the neutral axis is a straight line; iii) only small deflections are considered and the end point of the beam moves along a straight line orthogonal to the undeformed axis of the beam. In order to reduce the model complexity, it is usually assumed that the shear does not give contribution to the elastic energy and only the effect of the bending moment is considered; the rotational energy could not be considered in case of slender structures with small deflections; in case of beams moving in three dimensions, there is no coupling between the motion in orthogonal directions. The model employed to design the control law must be carefully chosen, since in some case these hypothesis are not verified. For instance, iii) is not verified for very slender structures like fishing-rods.

2.2 Exact Model

When all the above reported hypotheses are verified, the Euler Bernoulli equation of the single link can be derived:

$$\frac{\partial^2}{\partial x^2} \left[EJ \frac{\partial^2 w(x, t)}{\partial x^2} \right] = -\rho \frac{\partial^2 w(x, t)}{\partial t^2} + p(x, t) \quad (2.1)$$

where $w(x, t)$ is the deflection at abscissa x and time t , $p(x, t)$ is the external load, ρ is the mass per unit length, E is the Young's modulus and J is the cross section momentum of inertia. This result can be obtained using the Hamilton principle. The same results can be obviously achieved using the Lagrange equation directly [44]. If the cross section inertia and the Young modulus are constant along the beam, it follows:

$$EJ \frac{\partial^4 w(x, t)}{\partial x^4} + \rho \frac{\partial^2 w(x, t)}{\partial t^2} = p(x, t). \quad (2.2)$$

Damping is not taken into account but it can be introduced by means of $p(x, t)$ assuming a proper law. If no external loads act on the beam, the equation admits at least two kinds of solutions.

Traveling waves — A possible solution is

$$w(x, t) = f(kx \pm \omega t) = \sin(kx \pm \omega t) \quad (2.3)$$

where k and ω can be determined substituting in (2.2). For instance, if the beam has length L and is hinged at the extremities, it follows:

$$v = n \frac{\pi}{L} \sqrt{\frac{EJ}{\rho}}. \quad (2.4)$$

The higher the frequency, the higher is the wave speed. This particular solution should be considered for slender structures, when the propagation delay is significant.

The (differential) eigenvalues problem — The following stationary solution is considered

$$w(x, t) = \phi(x) \psi(t). \quad (2.5)$$

Substituting in (2.2) yields

$$EJ \psi(t) \frac{\partial^4 \phi(x)}{\partial x^4} + \rho \phi(x) \frac{\partial^2 \psi(t)}{\partial t^2} = 0 \quad (2.6)$$

The solution for the eigenvalues problem is

$$\psi(t) = A \sin(\omega t) + B \cos(\omega t) \quad (2.7)$$

with the following eigenfunction(s) ($\beta^4 = \omega^2 \rho / EJ$):

$$\phi(x) = c_1 \sin(\beta x) + c_2 \cos(\beta x) + c_3 \sinh(\beta x) + c_4 \cosh(\beta x). \quad (2.8)$$

The constants can be determined assuming the proper boundary conditions. The eigenfunctions are orthogonal with respect to the inner product and can be normalised:

$$\int_0^L \phi_i(x) \phi_j(x) dx = 0 \quad i \neq j \quad \int_0^L \phi_i(x) \phi_i(x) dx = 1. \quad (2.9)$$

In case of a clamped beam the boundary conditions are (deflection and rotation at the clamped extremity, shear and moment at the other)

$$w(0, t) = 0 \quad \frac{dw(0, t)}{dx} = 0 \quad EJ \frac{d^3 w(L, t)}{dx^3} = 0 \quad EJ \frac{d^2 w(L, t)}{dx^2} = 0 \quad \forall t. \quad (2.10)$$

These boundary conditions are verified if the following characteristic equation holds:

$$\cos(\beta L) \cosh(\beta L) = -1 \quad (2.11)$$

which is satisfied by (infinite and numerable) β_i , determining the natural frequencies:

$$\omega_i = \beta_i^2 \sqrt{\frac{EJ}{\rho}}. \quad (2.12)$$

In case of a hinged beam the boundary conditions are (deflection and moment at the hinge, shear and moment at the free extremity)

$$w(0, t) = 0 \quad EJ \frac{d^2 w(0, t)}{dx^2} = 0 \quad EJ \frac{d^3 w(L, t)}{dx^3} = 0 \quad EJ \frac{d^2 w(L, t)}{dx^2} = 0 \quad \forall t \quad (2.13)$$

with the following characteristic equation:

$$\sin(\beta L) \cosh(\beta L) = \sinh(\beta L) \cos(\beta L). \quad (2.14)$$

which gives the natural frequencies, different from (2.11). The normalised eigenfunctions can be determined using the second of (2.9) and (2.10) or (2.13). Further details can be found in some classical books [44], [43], [59] or specific articles, e.g. [9], [29], [55].

2.3 Rayleigh-Ritz Methods

When the solution to the eigenvalue problem is not feasible or the closed form solution cannot be computed, approximate solutions can be obtained by means of the Rayleigh-Ritz methods. These methods assume a certain function to describe the shape of the neutral axis of the link. The following ones could be employed:

Eigenfunctions — They must satisfy both the differential equations and the boundary conditions;

Comparison functions — They are 4 times differentiable, satisfy all the boundary conditions but not necessary the differential equations; this class includes the eigenfunctions;

Admissible functions — They are 2 times differentiable and satisfy only the geometrical boundary conditions; this class includes the comparison functions.

The geometric boundary conditions reflect the geometric constraints; the natural ones reflect the constraints on forces and moments [44].

In the Rayleigh-Ritz method, the comparison functions of a complete-in-energy set should be employed. If the interest is focused in the approximate solution of the eigenvalue problem, the solution can be found in the space of the admissible solutions instead of the comparison functions.

A series of polynomial can be employed, which can be properly truncated to a certain order. These solutions satisfy all the geometric boundary conditions but not the differential equations.

$$w(x, t) = \sum_{i=0}^{\infty} \psi_i(t) x^i. \quad (2.15)$$

A suitable truncation is performed to determine the functions $\psi_i(t)$ assuming the proper geometric boundary conditions. Further details can be found in [46], [56]. It can be proved that, notwithstanding this approximation does not verify the differential equations of motion, the solution converges to the exact one as the order of the truncation increases [27].

2.4 Finite Element Method

The finite element method can be regarded as a Rayleigh-Ritz method; instead of global admissible solutions extended over the entire domain (length) of the beam, the admissible functions are defined in subdomains (elements of the beam) and are usually low-degree polynomials [44], [43], [59].

The i -th element of length L_i is characterised by the rotations and displacements at the extremities:

$$\Psi_i(t) = [z_i \quad L_i\theta_i \quad z_{i+1} \quad L_i\theta_{i+1}]^T \quad (2.16)$$

In case of bending vibrations, the Hermite cubics can be employed as interpolation functions; the functions in the i -th subdomain are

$$\Phi_i(x) = [3\xi^2 - 2\xi^3 \quad \xi^2 - \xi^3 \quad 1 - 3\xi^2 + 3\xi^3 \quad -\xi + 2\xi^2 - \xi^3]^T \quad (2.17)$$

where $\xi = x_i/L$; $x_i \in [0, L_i]$. The admissible function in the i -th subdomain is

$$w_i(x, t) = \Phi_i^T(x) \Psi_i(t) \quad (2.18)$$

This function allows to compute the kinetic and elastic energy of the element:

$$T_i(t) = \frac{1}{2} \int_0^L \rho(x) \left(\frac{\partial w_i(x, t)}{\partial t} \right)^2 dx = \dot{\Psi}_i^T \frac{M_i}{2} \dot{\Psi}_i. \quad (2.19)$$

$$U_i(t) = \frac{1}{2} \int_0^L EJ(x) \left(\frac{\partial^2 w_i(x, t)}{\partial x^2} \right)^2 dx = \Psi_i^T \frac{K_i}{2} \Psi_i \quad (2.20)$$

Matrices M_i and K_i are constant positive definite and depend on the geometrical and mechanical parameters of the beam. The internal geometric boundary conditions are satisfied by imposing the congruence of the displacements and rotations at the element extremities. In practice the equations of the whole structure are obtained by a proper assembly of submatrices M_i and K_i .

The method can be extended also to three dimensional cases and to trusses and frames. The hierarchical finite elements method combines the advantages of both the Rayleigh-Ritz and the finite element methods [43].

2.5 Multiple Links

The exact model of a robot with a rigid link and a flexible forearm can be derived referring to the Hamilton principle, including the kinetic energy of the rigid link and the kinetic and potential energies of the flexible one. This approach can be generalised also to more complex structures, but the results are quite involved and are difficult to be employed for control design purposes. The model is described by a system of non linear partial differential

equations. An example is reported in [1]. Approximate models can be derived by means of the Rayleigh-Ritz method [58]. In the paper the authors develop the model of the rigid robot; the last link is modelled using the clamped-free eigenfunctions.

The extension to multiple flexible link is plain, in theory. An example is reported in [11], where an automatic symbolic modelling method is introduced. In [15] the dynamics of multi-link spatial manipulators with flexible links and joints is modelled by a redundant Lagrangian/finite element approach. The elastic deformations of the links are expressed in their tangential local floating frames. Both the rigid and elastic degrees of freedom of the system are treated as generalised coordinates, taking into account the coupling between rigid body and elastic motion. The constraint equations, representing kinematical relations among different coordinates due to connectivity of the links, are added to the equations of motion of the system by using Lagrange multipliers. The model includes nonlinear ordinary differential equations and nonlinear algebraic equations depending on the coordinates and Lagrange multipliers. It can be converted into a set of differential equations, which is solved numerically to predict the dynamic behaviour of the system.

Finally, a finite element approach can always determine a model almost automatically, also for three dimensional structures. The drawback is the huge dimension of the model. On the other hand a proper truncation could achieve smaller dimensions; in particular, examining the single body of the structure could give rise to suitable approximation also before the assembly process.

2.6 Remarks

The exact solution is useful to compute the “true” vibration frequencies of the beam. In control application it is usually approximated by some eigenfunctions, derived assuming suitable boundary conditions. It should be noted that in case of multiple links, the natural boundary conditions depends on the control. The Rayleigh-Ritz methods allows dealing only with geometrical boundary conditions, which depend on the configuration of the structure. For instance when two links are orthogonal, the joint can be considered a hinge; when the links are parallel, since deflection is possible, the hinge constraint is only an approximation, and the constrained extremities tend to be free as the stiffness of the first link decreases. The finite element approach can deal with all these cases; on the other hand, a large number of elements could be necessary to approximate the natural modes, giving rise to large order model. The model order can be reduced by means of low frequency approximations, but the employed algorithm should be carefully chosen, since the linearised model has the eigenvalues near the imaginary axis and could be non minimum phase.

3 Control

3.1 PD Regulation

The basis in control of robots with flexible links is the regulation of an equilibrium point. It can be shown that the decentralised PD regulation of the motors can achieve the asymptotic stability of the equilibrium point. The efficiency of the algorithm (i.e. the damping that can be introduced), depends on the coupling between the motors and the link: only if the vibration of the flexible link causes the motors to move, damping can be achieved; therefore direct drive motors with colocated sensors are taken into account. Some papers are devoted to show that a PD decentralised control law can impose and asymptotically stabilise a unique equilibrium point. In a first approach, reference is made to a two link robots with the last one flexible. The model includes the equations of the flexible beam in the Euler Bernoulli form and neither discretisation nor truncation are required. Assuming that no gravity is present, the global asymptotic stability of the unique equilibrium point can be proved by means of the Lyapunov method. The candidate Lyapunov function includes also the elastic and kinetic energy of the flexible beam and other terms related to the rigid motion. The LaSalle-Krasowskii theorem can be employed to prove asymptotic stability. The proof does not depend on the particular discretisation/truncation of the model equations and therefore is not sensitive to spillover. The result is also robust with respect to parameter uncertainties [1]. Only motor speed and position are required, but the proof can be easily extended to the case in which the PD regulator is replaced by a lead compensator; in this case only the motor position is necessary. In case of gravity a compensation term is introduced. The proof is reported in [14] where the damped and undamped case are examined. The compensation term can be computed either in the desired final position or along the trajectory. Since no feedback exists on the tip position error and only motor position and speed are employed, the payload and the masses must be known to achieve the desired point.

3.2 Strain Feedback Control

This technique is also employed in the regulation and includes in the control law some terms that depend on the strain of some points of the flexible links [41], [49], [28].

3.3 Inverse Dynamics

In case of trajectory tracking, the inverse kinematics and/or dynamics of flexible robots is required. Some algorithms have been investigated. References can be found in [5], [8], [10].

3.4 Fuzzy Regulation

A colocated decentralised PD regulation can achieve global asymptotic stability of the equilibrium point. On the other hand it is not possible to impose an arbitrary damping to all the modes. To improve the damping ratio and reduce the overshoot, it is possible to modulate the gains of the PD regulator in function of some outputs. A careful investigation of the behaviour of the plant can carry to the definition of rules to adapt the PD gains [17], [18], but the results strictly depend on the depth of the physical insight. The main property of this approach is that asymptotic stability can be demonstrated.

The application to multi link flexible robots requires a huge set of rules and membership functions [45]. The problem can be overcome introducing an adaptation algorithm to synthesise and tune the rules on-line. Acceleration feedback of the end effector is required. The adaptation scheme is of MRAS kind. Wide explanations can be found in [45], [34], where the method is applied to a two flexible link robot. Another application can be found in [25], where acceleration feedback is also required. Other references to fuzzy learning control are [37], [35], [36].

3.5 Robust Control

H_∞ techniques are employed to design robust controllers for flexible links with uncertain flexible dynamics. If it is assumed that there is not damping, the link has all the eigenvalues on the imaginary axis, and therefore the necessary conditions to design an H_∞ controller are not satisfied. Safonov [48] introduces a right shifting transformation in the s-plan to make the poles with real positive part. Since the H_∞ design does not cancel unstable poles, the controller is stable and remains stable after the inverse transformation. In practice the right-shifting allows only poor performance. Left-shifting is more feasible; on the other hand, since the regulator cancels the stable poles of the shifted plant, it will have imaginary zeros. Another approach consists in adopting a High Authority/Low Authority Control architecture: an inner loop makes the controlled system asymptotically stable, even if with poor performance (Low Authority Control); the external loop is designed referring to a more suitable system, in order to achieve the control system specifications (High Authority Control). Safonov [48] uses in the inner loop a PD regulator; Banavar [7] uses LQG controllers. The first approach employs a colocated PD regulator that is robust with respect to model uncertainties; the other method gives rise to better performance, because it implements a state feedback, but it is less robust in presence of relevant uncertainties; moreover it is sensitive to spillover; anyway, experiments show that the LQR regulator is effective. The H_∞ controller can be designed using the Mixed Sensitivity Approach. A drawback consists in the fact that the whole system does not ensure zero error for step input. This is implicit in the Glover Doyle theorem, which requires that the input must be L2 integrable. A feedforward term could be employed,

but since it introduces implicitly a pole in the origin, the stabilising effect of the inner loop is lost. In practice the feedforward action can only reduce the steady state error.

Another technique is named combined pole placement/sensitivity function shaping method [33], [32]. Reference is made to a discrete time representation of the flexible link, the model of which is suitably truncated to represent the significant dynamics. After choosing the sensitivity and the complementary sensitivity functions, an iterative procedure is employed to design the controller, achieving the desired specifications.

The μ -techniques have also been applied. In this case, since structured uncertainties are taken into account, the controller is less conservative than an H_∞ one. An example of application can be found in [30]. Improvements of the μ -techniques, developed for structural control applications, can be found in [60], where a mixed H_2/H_∞ technique is employed. Here, a set of mixed H_2/H_∞ compensators are designed, which are optimised for a fixed compensator dimension. The mixed norm recovers the H_2 design performance levels, while providing the same level of robust stability as in the μ design.

3.6 Singular Perturbation

When the robot dynamics can be divided in fast and slow, the singular perturbation approach can be employed [20]. Firstly the robot is partially feedback linearised [52], improving also the separation between the slow and fast dynamics. The fast dynamics depends on the flexibility and is considered as a perturbation of the slow dynamics, which coincides with the rigid motion. The controller is composed of a slow term that ensures tracking, and a fast term that damps the vibrations. Exponential stability can be demonstrated. A preliminary requisite is that the bandwidth for tracking must be well distinguished from the vibration modes; from another point of view, either the robots is rather rigid or the required bandwidth for trajectory tracking must be small. Some references can be found in [4], where the technique is experimentally applied to a single flexible link. Applications to multilink robots can be found in [3], [64].

3.7 Sliding

The standard sliding mode technique requires to define a sliding surface for each Lagrangian coordinate; to drive the trajectories on the sliding surface the same number of control inputs are required. In case of flexible links, where there are more Lagrangian coordinates than inputs, it is required to adopt combined sliding surfaces. It is necessary to show that the equivalent motion converges to zero. This is easy in case of single arms [61], [62] or multilink robots, if the coupling between the links is negligible [12]. In case of two link robots with a rigid and a flexible link, it can be shown the dynamics on the sliding surfaces is linear and can be stabilised by a suitable

choice of the parameters of the sliding surfaces [16]. The control law requires some deflection measurements on the flexible link. The control chattering is avoided approximating the ideal relay with a high gain saturated amplifier. Experimental tests can be found in [4]. Another approach is the continuous sliding-mode [63], which can be applied to minimum phase system. An example can be found in [26].

3.8 Input Shaping

Shaped commands profiles are generated by convoluting a sequence of impulses with the desired command signal. One of the problem to be solved is the insensitivity, that is the ability of the input shaper to reduce the residual vibrations in presence of modelling error. Details and comparisons of some techniques are reported in [54]. Experimental results of the shaping technique are reported in [31], [13].

3.9 Cyclic Control

This approach is useful when the motion occurs between some equilibrium points and the whole path is not specified [39], [40]. Moreover, the problem of trajectory tracking for a flexible arm between two equilibrium points is made more difficult by the non-minimum-phase behaviour of this system. The problem can be classified as a state steering one. On the basis of these motivations, the problem of steering the state of a control system by learning has been investigated theoretically. The algorithms are based upon the selection of a finite-dimensional subspace of the linear space or all control functions defined over a finite time interval. As the search for the steering control is restricted to this subspace, the resulting algorithm is finite dimensional [38].

3.10 Predictive Control

Since in many applications the most of the vibration energy is not generated by disturbances but by the actuators [31], great advantages can be obtained by a feedforward action which, by suitable shaping of the input command, can improve the performance of an existing feedback controller. Feedforward controllers can be designed to implement an inverse model control scheme, aiming at the cancellation of the known dynamics of the open or closed loop system [51]; in this way, if a reliable model is available, perfect tracking of any trajectory can be obtained. The drawback of this approach is the difficulty in the design of general model inversion algorithms for non linear systems and in the lack of robustness. A problem which arises in the design of a feedforward inverse filter for linear flexible links is related to the fact that the transfer function between the motor and the tip is non minimum phase, implying an unstable inverse controller. This fact can be partially overcome by

designing a stable approximation of the inverse model [57]. Another method for the approximated model inversion is the Model Predictive Control approach [24]. Model Predictive Controllers (MPCs) plan on-line a suitable sequence of future input commands on the basis of a prediction model, in order to track a desired output trajectory and minimising a defined index of performance. Although MPCs have been mainly used in process control, they have been recently applied also to the control of mechanical systems characterised by non minimum-phase transfer functions and with badly damped poles [50], [2]. Another attractive feature of MPCs consists of their ability to handle constraints on control signals and state variables; these aspects are quite important because it is possible to generate sophisticated optimal control laws satisfying multi-objective constrained performance criteria. Closed form MPCs are available only for linear systems minimising a quadratic cost function. Today analytical solutions are not available for general nonlinear MPC and efficient numerical procedures exist only for convex optimisation problems. In several papers Lagrange operators and quadratic programming have been used to manage nonlinear constrained optimisations.

In MPC the optimal command sequence is determined as the result of an optimisation problem at each sampling instant. All the performance specifications are quantified by means of a cost index. An index, which is often employed, is:

$$J = P \sum_{j=N_1}^{N_2} \|\hat{y}(k+j|k) - y_d(k+j)\|_P^2 + Q \sum_{j=1}^{N_c} \|\Delta u(k+j-1)\|_Q^2 + J_1. \quad (3.1)$$

The first term of J evaluates the square error between the desired future output signal $y_d(k+j)$ and the predicted future output $\hat{y}(k+j|k)$, estimated on the basis of the prediction model and the feedback information available at instant k . This error is evaluated over a defined prediction horizon, which is a window of $N_2 - N_1$ samples. The second term of J weights the control effort of the sequence of the input increments $\Delta(k) = u(k) - u(k-1)$ over the control horizon window of N_u samples. The diagonal matrices P and Q comprise weight coefficients, which are used to give a different emphasis to the terms of J . The term J_1 is a further cost function, which can be used to take into account other specifications or constraints. MPC is mainly used in conjunction with constraints on the amplitude and the rate of variation of the input and output variables. The minimisation of index J is performed with regard to the sequence of the control increments $[\Delta u(k), \Delta u(k+1), \dots, \Delta u(k+N_u-1)]$, $j = 1, \dots, N_U$.

A key aspect in predictive control is the application of the receding horizon strategy. In this approach only the first sample of the optimal sequence is applied to the system; subsequently the horizon is moved one step in the future and the optimisation is repeated on the basis of the measured feedback information.

The computational load required by classical optimisation algorithms with a great number of decision variables is the main difficulty encountered in non-linear constrained MPC design. A way to overcome these problems is to fuse a MPC scheme with intelligent control optimisation techniques. In this context a very promising approach is to exploit Evolutionary Algorithms (EAs). EAs are pseudo random parallel search algorithms, the inference engine of which is based on the principle of biological evolution. One of the main advantages of evolutionary optimisation is the only linear increase of computational time with the number of decision variables; the main disadvantage, when employed in feedback control, is the lack of results proving the stability of the control law; on the other hand EAs showed to be quite appropriate optimisers for MPC [47], [42]. In [21] an Evolutionary MPC (EMPC) is used as feedforward controller in order to improve the trajectory tracking accuracy of a feedback controlled mechanical system. In [22] a $PD+H_\infty$ controller has been employed (Figure 3.1), while in [23] only a PD regulator. The aim of the feedback loop is to achieve the asymptotic stability even if with poor performance. To guarantee a certain robustness, some information from the plant can be employed (realignment): the actual output error can be fed back and included in the optimisation index; the nominal model used for the prediction can be simulated starting from the measured or estimated state of the plant; in this way the nominal model behaves like a one step predictor (Figure 3.2).

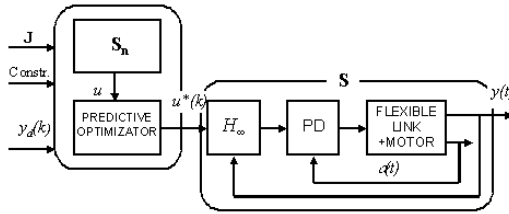


Figure 3.1. Pure Feedforward MPC

The complete freedom in the choice of the multivariable objective function and the possibility to handle constraints allow the fulfillment of additional performance of the whole control system.

As far as the real time implementation is concerned, in [23] some tests were carried out on an experimental set-up. In order to achieve the desired performance, the Evolutionary algorithm must be carefully tuned, with suitable genetic operators and parameters. An increase in the computational speed can be achieved by a proper coding of the decision variables, but a deeper insight in the optimisation problem and in the genetic algorithm can

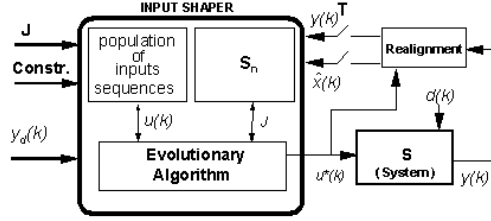


Figure 3.2. MPC with feedback information

give rise to a better performance. In particular, two improvements can be introduced.

Steady state reproduction mechanism during the optimisation for the k -th sample — Since a limited computation time is available, it is essential that the good solutions founded during previous evolutions are not lost, but are used as “hot starters” for the next generation. To implement this strategy, the best chromosomes pass unchanged from the current generation to the next one, ensuring a not decreasing fitness function for the best individual of the population. The remaining part of the population is generated on the basis of a rank selection mechanism: a certain number of the best individuals constitute a mating pool; within the mating pool two random parents are selected and two children are created applying crossover and mutation operators; this operation lasts until the new population is entirely generated.

Heredity from time interval k to $k+1$ — The results of optimisations at the k^{th} instant are not wasted, but they constitute part of the starting population for the next time step optimisation. This approach can be easily applied in a receding horizon strategy; in fact the best chromosomes of the current generation computed at the end of the k -th time interval can be exploited as a privileged starting solution for the next time step optimisation. At the beginning of the $(k+1)$ -th time interval, since the horizon is shifted one step ahead, the N genes of the best solutions are shifted forward to the next location; in this way the first genes are lost and replaced by the second ones and so on; the values of the last positions are filled by keeping unaltered the corresponding values of the previous optimisation. The remaining chromosomes are randomly generated. The application of hereditary information is of great relevance, because a dramatic improvement in the convergence speed of the algorithm can be achieved and a good updated solution can be found in few generations [22], [22], [23].

Some results are reported in the subsequent figures. Reference was made to a single flexible link. The controller must damp the oscillations when the link starts from a deformed condition. In Figure 3.3 the free response is reported and compared with a well tuned PD controller. In Figure 3.4 a pure

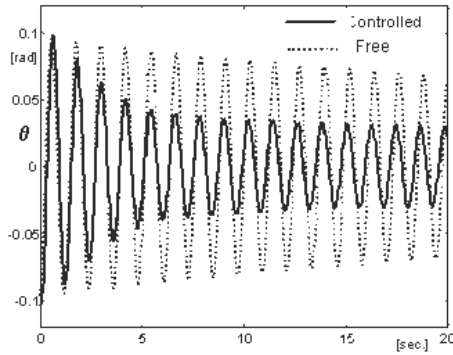


Figure 3.3. Free and PD controlled flexible link

feedforward MPC is employed. An increase in damping is achieved, but the mismatches between the nominal model employed for the prediction and the real plant cause some residual oscillations. In fact the prediction model is controlled to zero, contrary to the real plant. Finally, in 3.5 an intermittent realignment is introduced; the residual oscillation are very small, but they are not disappeared; this effect depends on the evolutionary algorithm that cannot generate a zero command signal, because of the quantisation. A further improvement consists in introducing an adaptation mechanism, which reduce the quantisation effects as the error decrease, causing the residual oscillations to vanish.

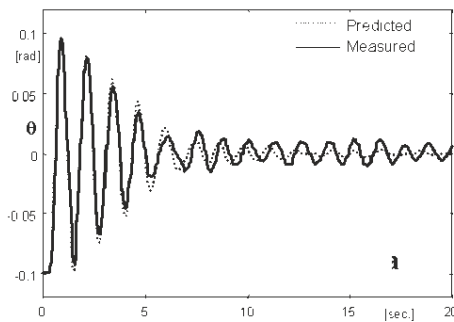


Figure 3.4. Flexible link with pure feedforward MPC

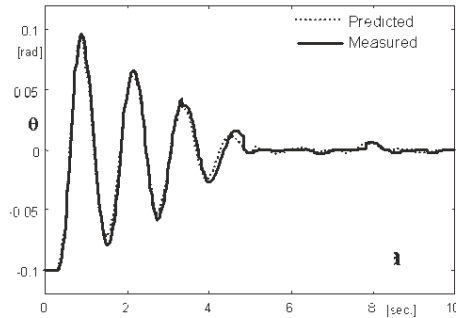


Figure 3.5. Flexible link with intermittent realignment

4 Conclusion

A brief survey on control of flexible robots has been reported. The literature about these arguments is so huge that only some citations have been included. For a deeper insight, reference could be made to the bibliography hereafter.

References

1. Ahmet A, Yigit S 1994 On the stability of PD control for a two link rigid flexible manipulator. *ASME Journal of Dynamic Systems, Measurement, and Control* 116:208–215
2. Alvarez T, Tadeo F, de Prada C 1998 Constrained predictive control of multi-variable robotic systems. In: *Proceedings of 1998 IEEE Conference on Control and Applications* Trieste, I, pp 588–592
3. Aoustin Y, Chevallereau C 1993 The singular perturbation control of a two flexible links robot. In: *Proceedings of 1993 IEEE International Conference on Robotics and Automation* Atlanta, GA, pp 3:737–742
4. Aoustin Y, Chevallereau C, Glumineau A, Moog C H 1994 Experimental results for the end-effector control of a single flexible robotic arm. *IEEE Transactions on Control Systems Technology* 2:371–381
5. Asada H, Zae Z D, Tokumaru H 1990 Inverse dynamics of flexible robots arms: Modeling and computation for trajectory control. *ASME Journal of Dynamic Systems, Measurement and Control* 112:177–185
6. Bäck T, Schwefel H P 1996 Evolutionary computation: An overview. In: *Proceedings of IEEE International Conference on Evolutionary Computation* Nagoya, J, pp 20–29
7. Banavar R N, Dominic P 1995 An LQG/Hinf controller for a flexible manipulator. *IEEE Transactions on Control Systems Technology* 3:409–416
8. Bayo E, Papadopoulos P, Stubbe J 1989 Inverse dynamics and kinematics of multi link elastic robots: an iterative frequency domain approach. *International Journal of Robotics Research* 8(6):49–62

9. Bellezza F, Lanari L, Ulivi G 1990 Exact modeling of the flexible slewing link. *Proceedings of 1990 IEEE International Conference on Robotics and Automation* Cincinnati, OH, pp 734–739
10. Carrera E, Serna M A 1996 Inverse dynamics of flexible robots. *Mathematics and Computer in Simulations* 41:485–508
11. Cetikunt S, Ittoop B 1992 Computer-automated symbolic modelling of dynamics of robotic manipulators with flexible links. *IEEE Transactions on Robotics and Automation* 8:94–105
12. Chen Y P, Yeung K S 1991 Sliding mode control of multi-link flexible manipulators. *International Journal of Control* 54:257–278
13. Cho J K, Park Y S 1996 Experimental evaluation of time-varying impulse shaping with a two-link flexible manipulator. *Robotica* 14:339–345
14. De Luca A, Siciliano B 1993 Regulation of flexible arms under gravity. *IEEE Transactions on Robotics and Automation* 9:463–467
15. Farid M, Lukasiewicz S A 2000 Dynamic modeling of spatial manipulators with flexible links and joints. *Computers and Structures* 75:419–437
16. Ficola A, La Cava M 1996 A sliding mode controller for a two joint robot with an elastic link. *Mathematics and Computers in Simulations* 41:559–569
17. Ficola A, La Cava M 1996 A class of fuzzy regulators for vibration damping in flexible structures. In: *Proceedings of IASTED International Conference on Robotics and Manufacturing* Honolulu, HI, pp 255–260
18. Ficola A, La Cava M 1996 A class of fuzzy regulators for vibration control in flexible structures. In: *Proceedings of 4th International Conference on Control, Automation, Robotics, and Vision* Singapore, pp 2180–2184
19. Foegel B 1994 Applying evolutionary programming to selected control problems. *Computers Mathematical Applications* 27(11):89–104
20. Fraser A R, Daniel R W 1991 *Perturbation Techniques for Flexible Manipulators* Kluwer, Norwell, MA
21. Fravolini M L, Ficola A, La Cava M 1999 Improving trajectory tracking by feedforward evolutionary multivariable constrained optimization. In: *Proceedings of 1999 IEEE/ASME International Conference on Advanced Intelligent Mechatronics* Atlanta, GA, pp 985–990
22. Fravolini M L, Ficola A, La Cava M 1999 Vibration reduction by predictive evolutionary trajectory shaping. In: *Proceedings of 38th IEEE Conference on Decision and Control* Phoenix, AZ, pp 5289–5291
23. Fravolini M L, Ficola A, La Cava M 2000 Intermittent feedback strategies for the predictive control of a flexible link: experimental results. In: *Preprints of 6th IFAC Symposium on Robot Control* Vienna, A, pp 391–396
24. Garcia C E, Prett D M, Morari M 1989 Model predictive control: theory and practice. A survey. *Automatica* 25:335–348
25. Green A, Sasiadek J Z 2000 Direct and fuzzy tracking control of a two link flexible manipulator. In: *Preprints of 6th IFAC Symposium on Robot Control* Vienna, A, pp 555–560
26. Hisseine D, Lohmann B 2000 Sliding mode tracking control for a single-link flexible robot arm. In: *Preprints of 6th IFAC Symposium on Robot Control* Vienna, A, pp 579–584
27. Indri M, Tornambè A 1996 Lyapunov analysis of the approximate motion equations of flexible structures. *Systems and Control Letters* 28:31–41
28. Junkins J L, Rahman Z, Bang H 1992 Near-minimum-time control of distributed parameter systems: analytical and experimental results. *AIAA Journal of Guidance, Control and Dynamics* 14:406–415
29. Kanoh H 1991 Distributed parameter models of flexible robot arms. *Advanced Robotics* 5, 87–99

30. Karkoub M, Balas G, Tamma K, Donath M 2000 Robust control of flexible manipulators via μ -synthesis. *Control Engineering Practice* 8:725–734
31. Khorrami F, Jain S, Tzes A 1995 Experimental results on adaptive nonlinear control and input preshaping for multilink flexible manipulators. *Automatica* 31:83–97
32. Landau I D 1995 Robust digital control of systems with time delay. *European Journal of Control* 62:325–347
33. Landau I D, Langer J, Rey D, Barnier J 1996 Robust control of a 360deg flexible arm using the combined pole placement/sensitivity function shaping method. *IEEE Transactions on Control Systems Technology* 4:369–383
34. Layne J R 1992 *Fuzzy Model Reference Learning Control* MS Thesis, Ohio State University
35. Layne J R, Passino K M 1992 Fuzzy model reference learning Control. In: *Proceedings of 1st IEEE Conference on Control Applications* Dayton OH, pp 686–691
36. Layne J R, Passino K M, Yurkovich S 1993 Fuzzy learning control for anti skid braking systems. *IEEE Transactions on Control Systems Technology* 1:122–129
37. Lee C 1990 Fuzzy logic in control system: fuzzy logic controllers. *IEEE Transactions on Systems, Man and Cybernetics* 2:404–418
38. Lucibello P 1992 Output regulation of flexible mechanisms by learning. In: *Proceedings of 31st IEEE Conference on Decision and Control* Tucson, AZ, pp 2533–2538
39. Lucibello P, Panzieri S 1998 Application of cyclic control to a two link flexible arm. *Automatica* 34:1025–1029
40. Lucibello P, Panzieri S, Ulivi G 1997 Repositioning control of a two link flexible arm by learning. *Automatica* 33:579–590
41. Luo Z H 1993 Direct strain feedback control of flexible robot arms: New theoretical and experimental results. *IEEE Transactions on Automatic Control* 38:1610–1622
42. Martinez M, Senent J, Blasco X 1998 Generalized predictive control using genetic algorithms. *Engineering Application of Artificial Intelligence* 11:355–367
43. Meirovitch L 1986 *Elements of Vibration Analysis* McGraw-Hill, New York, NY
44. Meirovitch L 1990 *Dynamics and Control of Structures* Wiley, New York, NY
45. Moudgal V G 1993 *Direct, Supervisory and Adaptive Fuzzy Control of a Two Link Flexible Manipulator* MS Thesis, Ohio State University
46. Nicosia S, Tomei P, Tornambè A 1989 Hamiltonian description and dynamic control of a flexible arm. *Journal of Robotic Systems* 6:345–361
47. Onnen C, Babuska R, Kaymak U, Sousa J M, Verbruggen H B, Iserman R 1997 Genetic algorithms for optimization in predictive control. *Control Engineering Practice* 5:1363–1372
48. Safonov M G, Chiang R Y, Flashner H 1991 Hinf control synthesis for a large space structure. *AIAA Journal of Guidance, Control and Dynamics* 14:513–520
49. Sakawa Y, Luo Z H 1989 Modeling and control of coupled bending and torsional vibrations of flexible beams. *IEEE Transactions on Automatic Control* 34:970–977
50. Sepehri N, Wu G 1998 Experimental evaluation of generalized predictive control applied to a hydraulic actuator. *Robotica* 16:463–474
51. Shiller Z, Chang H 1995 Trajectory preshaping for high speed articulated systems. *ASME Journal of Dynamic Systems, Measurement and Control* 117:304–310

52. Siciliano B, Book W J 1988 A singular perturbation approach to control of lightweight flexible manipulators. *International Journal of Robotics Research* 7(4):79–90
53. Singer N, Seering W 1992 An extension of command shaping method for controlling residual vibration using frequency sampling. In: *Proceedings of 1992 IEEE International Conference on Robotics and Automation* Nice, F, pp 800–805
54. Singhose W, Pao L 1997 A comparison of input shaping and time-optimal flexible-body control. *Control Engineering Practice* 5:459–467
55. Tokhi M O, Azad A K M 1996 Modeling of a single link flexible manipulator system: theoretical and practical investigation. *Robotica* 14:91–102
56. Tomei P, Tornambè A 1988 Approximate modeling of robots having elastic links. *IEEE Transactions on Systems, Man and Cybernetics* 18:831–840
57. Torfs E, Vuerinckx R, Swevers J, Schoukens J 1998 Comparison of two feedforward design method aiming at accurate trajectories tracking of the end point of a flexible link. *IEEE Transactions on Control Systems Technology* 6:2–14
58. Wang D, Vidyasagar M 1992 Modeling a class of multilink manipulators with the last link flexible. *IEEE Transactions on Robotics and Automation* 8:33–41
59. Weaver W, Timoshenko S P, Young D H 1990 *Vibration Problems in Engineering* Wiley, New York, NY
60. Worthon M, Calise A J, 1996 Homothopy algorithm for fixed order mixed H2/Hinf Design. *AIAA Journal of Guidance Control and Dynamics* 19:1262–1269
61. Yeung K S, Chen Y P 1989 Regulation of a one-link flexible robot arm using sliding mode technique. *International Journal of Control* 49:1965–1978
62. Yeung K S, Chen Y P 1990 Sliding mode controller design of a single-link flexible manipulator under gravity. *International Journal of Control* 52:101–117
63. Zhou F, Fisher D G 1992 Continuous sliding mode control. *International Journal of Control* 55:313–327
64. Zhu S Q, Commuri S, Lewis F L 1994 A singular perturbation approach to stabilization of the internal dynamics of multilink flexible robots. In: *Proceedings of 1994 American Control Conference* Baltimore, MD, pp 1386–1390

Interaction Control

Fabrizio Caccavale, Ciro Natale, Bruno Siciliano, and Luigi Villani

PRISMA Lab, Dipartimento di Informatica e Sistemistica
Università degli Studi di Napoli Federico II
<http://www.prisma.unina.it>

In the framework of interaction control of robotic systems, impedance control represents one of the most effective strategies when the model of the environment is not available. In this chapter, a new impedance control strategy is presented for six-degree-of freedom (six-DOF) tasks. The main features are geometric consistency and absence of singularities. The case of a single manipulator interacting with the environment is considered first. Then, the case of redundant manipulators is analysed, and an algorithm ensuring stabilisation of null-space velocities is presented. The redundant degrees of mobility are exploited to optimise an additional task function. Finally, the case of cooperative robots manipulating a common object is addressed: both the problems of loose and tight cooperation are considered. The theoretical findings are validated in experiments on a dual-robot industrial setup.

1 Introduction

Control of interaction between a robot manipulator and the environment is crucial for successful execution of a number of practical tasks where the robot end effector has to manipulate an object or perform some operation on a surface. Typical examples include polishing, deburring, machining or assembly. During interaction, the environment sets constraints on the geometric paths that can be followed by the end effector. In such a case, the use of pure motion control for controlling interaction is a candidate to fail, while it is expected that enhanced performance can be achieved with an interaction control provided that force measurements are available [25].

Interaction control strategies can be grouped in two categories; those performing indirect force control and those performing direct force control. The main difference between the two categories is that in the former force control is achieved via motion control, without explicit closure of a force feedback loop; in the latter, instead, the contact force is controlled to a desired value, thanks to the closure of a force feedback loop.

To the first category belong compliance control [21], stiffness control [23] and also impedance control [14], where the position error is related to the contact force through a mechanical stiffness or impedance of adjustable parameters.

If a detailed model of the environment is available, a widely adopted strategy belonging to the second category is the hybrid position/force control [22],

[27], [17], [18]. In most practical situations, the model of the environment is not available; in such a case, effective strategies still in the second category are the inner/outer motion/force control [12] and the parallel control [10].

The focus of the present chapter is on impedance control for six-degree-of-freedom (six-DOF) tasks. Impedance control is aimed at imposing a desired dynamic behaviour to the end effector of a robot manipulator in the presence of external forces and moments, described by a mechanical impedance. For three-degree-of-freedom tasks (i.e. involving the sole end-effector position), the impedance consists in three linear second order differential equations corresponding to a mechanical system with desired mass, damping and stiffness. If a minimal representation of the end-effector orientation is adopted (e.g. Euler angles), the generalisation of the impedance to six-DOF tasks may result to be geometrically inconsistent; moreover, representation singularities may occur. In order to guarantee geometric consistency, the rotational part of the impedance equation is expressed in terms of a class of geometrically meaningful orientation representations based on the equivalent axis and angle of rotation. Singularities are avoided by using the unit quaternion, which is a four parameter angle/axis representation. An energy-based formulation is adopted to derive the impedance equation whose rotational part results to be nonlinear [7].

The geometrically consistent approach is applied also to the impedance control of redundant manipulators, characterised by a number of degrees of mobility greater than the number of degrees of freedom required for the execution of a given task. The method adopted for redundancy resolution allows complete decoupling of internal motion control from forces and moments acting on the end effector. Thus internal motion can be exploited to meet additional task requirements, e.g. to improve robot dexterity, without affecting or being affected by the execution of the interaction task [20].

In the framework of interaction control of cooperative robots manipulating a common object, both the problems of loose and tight cooperation are addressed.

A cooperative control strategy can be termed loose when the manipulation task is executed by controlling each robot in an independent fashion. Cooperation is realised only at the task planning level, e.g. mating rigid parts such as a dual-robot assembly in a workcell. In this case limited mating forces are achieved by adopting a geometrically consistent impedance control strategy for one of the two manipulators [5].

A cooperative control strategy can be termed tight when the manipulation task is executed by controlling the robots in a coordinated fashion. Cooperation is realised not only at the task planning level, but also at the control level. This is the typical task of two robots whose end effectors tightly grasp a commonly held rigid object, thus creating a closed-kinematic chain. In this case, interaction of the grasped object with the environment is successfully

managed by enforcing a geometrically consistent impedance behaviour at the object level [4].

The control strategies presented in this chapter have been experimentally tested on a set-up composed by two industrial robots Comau SMART-3S each of them equipped with a force/torque sensor ATI FT-30/100 mounted at the wrist. A number of experiments for representative robotic tasks is illustrated.

2 Motion Control

In this section the kinematic and dynamic modelling of a robot manipulator are briefly presented. Moreover, the motion control problem in the task space is analysed, which is the base for developing interaction control.

2.1 Modelling

The kinematic model of an open-chain robot manipulator gives the relationship between the $(n \times 1)$ vector of joint variables and the (3×1) position vector \mathbf{p}_e and the (3×3) rotation matrix \mathbf{R}_e , i.e. $\mathbf{p}_e = \mathbf{p}_e(\mathbf{q})$ and $\mathbf{R}_e = \mathbf{R}_e(\mathbf{q})$.

The quantities \mathbf{p}_e , \mathbf{R}_e characterise the end-effector frame $\Sigma_e(O_e-X_eY_eZ_e)$ with respect to a fixed base frame $\Sigma_b(O_b-X_bY_bZ_b)$ and no superscript is used; instead, if a matrix or vector quantity is to be referred to a frame other than the base frame, then a proper frame superscript shall precede the quantity.

Let $\dot{\mathbf{q}}$ denote the vector of joint velocities, $\dot{\mathbf{p}}_e$ the vector of end-effector linear velocity, and $\boldsymbol{\omega}_e$ the (3×1) vector of end-effector angular velocity. The differential kinematics model gives the relationship between $\dot{\mathbf{q}}$ and $\mathbf{v}_e = [\dot{\mathbf{p}}_e^T \ \boldsymbol{\omega}_e^T]^T$ in the form

$$\mathbf{v}_e = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}} \quad (2.1)$$

where \mathbf{J} is the $(6 \times n)$ end-effector geometric Jacobian matrix. A general motion task for the end-effector position and orientation requires m degrees of freedom with $m \leq 6$. Whenever the number of joints exceeds the number of degrees of freedom, i.e. $n > m$, the manipulator is said kinematically redundant.

The dynamic model can be written in the form

$$\mathbf{B}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{F}\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} - \mathbf{J}^T(\mathbf{q})\mathbf{h}, \quad (2.2)$$

where \mathbf{B} is the (6×6) symmetric positive definite inertia matrix, $\mathbf{C}\dot{\mathbf{q}}$ is the (6×1) vector of Coriolis and centrifugal torques, \mathbf{g} is the (6×1) vector of gravitational torques, $\boldsymbol{\tau}$ is the (6×1) vector of driving torques, $\mathbf{h} = [\mathbf{f}^T \ \boldsymbol{\mu}^T]^T$ is the (6×1) vector of contact force exerted by the end effector on the environment.

2.2 Motion Control in Task Space

The motion control problem for a robot manipulator can be formulated as finding the joint torques which ensure that the end effector attains a desired position and orientation. Since the present chapter is focused on the problem of controlling the interaction between the manipulator end effector and the environment, direct feedback of task-space variables (i.e. end-effector position and orientation) is utilised.

A classical control strategy is inverse dynamics control, which is aimed at linearising and decoupling the manipulator dynamics via feedback. The joint torques are chosen as

$$\boldsymbol{\tau} = \mathbf{B}(\mathbf{q})\boldsymbol{\alpha} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{F}\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) + \mathbf{J}^T(\mathbf{q})\mathbf{h} \quad (2.3)$$

where $\boldsymbol{\alpha}$ is a new control input to be properly designed.

Folding the control law (2.3) into the system model (2.2), and taking into account that $\mathbf{B}(\mathbf{q})$ is always nonsingular, yields

$$\ddot{\mathbf{q}} = \boldsymbol{\alpha} \quad (2.4)$$

which constitutes a linear and decoupled system, where the quantity $\boldsymbol{\alpha}$ represents a resolved acceleration in terms of joint variables.

The new control input $\boldsymbol{\alpha}$ in (2.4) can be chosen as

$$\boldsymbol{\alpha} = \mathbf{J}^{-1}(\mathbf{q}) \left(\mathbf{a} - \dot{\mathbf{J}}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} \right) \quad (2.5)$$

which leads to

$$\dot{\mathbf{v}}_e = \mathbf{a} \quad (2.6)$$

where the time derivative of (2.1) has been used. The vector \mathbf{a} attains the meaning of a resolved acceleration in terms of task space variables.

In deriving (2.5), a nonredundant manipulator ($n = 6$) moving in a singularity-free region of the workspace has been considered to compute the inverse of the Jacobian.

It is appropriate to partition the vector \mathbf{a} into its linear and angular components, i.e. $\mathbf{a} = [\mathbf{a}_p^T \ \mathbf{a}_o^T]^T$, where \mathbf{a}_p and \mathbf{a}_o are (3×1) vectors. Therefore, Equation (2.6) can be rewritten as

$$\ddot{\mathbf{p}}_e = \mathbf{a}_p \quad (2.7)$$

$$\dot{\boldsymbol{\omega}}_e = \mathbf{a}_o, \quad (2.8)$$

where \mathbf{a}_p and \mathbf{a}_o shall be designed so as to ensure tracking of the desired end-effector position and orientation trajectory, respectively.

The desired position trajectory is specified in terms of the position vector $\mathbf{p}_d(t)$, linear velocity vector $\dot{\mathbf{p}}_d(t)$ and linear acceleration vector $\ddot{\mathbf{p}}_d(t)$;

the desired orientation trajectory is specified in terms of the rotation matrix $\mathbf{R}_d(t)$, angular velocity vector $\boldsymbol{\omega}_d(t)$ and angular acceleration vector $\dot{\boldsymbol{\omega}}_d(t)$. The quantities \mathbf{p}_d and \mathbf{R}_d characterise the origin and the orientation of a desired frame Σ_d .

A position error between the desired and the actual end-effector position can be defined as $\Delta \mathbf{p}_{de} = \mathbf{p}_d - \mathbf{p}_e$ where the operator Δ denotes that a vector difference has been taken, and the double subscript denotes the corresponding frames. Then, the resolved linear acceleration can be chosen as

$$\mathbf{a}_p = \ddot{\mathbf{p}}_d + \mathbf{K}_{Dp} \Delta \dot{\mathbf{p}}_{de} + \mathbf{K}_{Pp} \Delta \mathbf{p}_{de} \quad (2.9)$$

ensuring exponential tracking of the desired position trajectory for any choice of the positive definite matrix gains \mathbf{K}_{Dp} and \mathbf{K}_{Pp} .

The most natural way of defining an orientation error is to consider an expression analogous to the position error, i.e. $\Delta \boldsymbol{\varphi}_{de} = \boldsymbol{\varphi}_d - \boldsymbol{\varphi}_e$, where $\boldsymbol{\varphi}_d$ and $\boldsymbol{\varphi}_e$ are the set of Euler angles that can be extracted respectively from the rotation matrices \mathbf{R}_d and \mathbf{R}_e describing the orientation of Σ_d and Σ_e .

The resolved angular acceleration based on the Euler angles error can be chosen as

$$\mathbf{a}_o = \mathbf{T}(\boldsymbol{\varphi}_e)(\ddot{\boldsymbol{\varphi}}_d + \mathbf{K}_{Do} \Delta \dot{\boldsymbol{\varphi}}_{de} + \mathbf{K}_{Po} \Delta \boldsymbol{\varphi}_{de}) + \dot{\mathbf{T}}(\boldsymbol{\varphi}_e, \dot{\boldsymbol{\varphi}}_e) \dot{\boldsymbol{\varphi}}_e \quad (2.10)$$

where \mathbf{T} is the transformation matrix relating the time derivative of the Euler angles $\boldsymbol{\varphi}_e$ and the end-effector angular velocity $\boldsymbol{\omega}_e$, i.e.

$$\boldsymbol{\omega}_e = \mathbf{T}(\boldsymbol{\varphi}_e) \dot{\boldsymbol{\varphi}}_e. \quad (2.11)$$

The control law (2.10) ensures exponential tracking of the desired orientation trajectory for any choice of the positive definite matrix gains \mathbf{K}_{Do} and \mathbf{K}_{Po} , provided that $\mathbf{T}(\boldsymbol{\varphi}_e)$ is nonsingular.

The above Euler angles error becomes ill-conditioned when the actual end-effector orientation $\boldsymbol{\varphi}_e$ becomes close to a representation singularity, i.e. a configuration for which $\mathbf{T}(\boldsymbol{\varphi}_e)$ becomes singular. In order to overcome this drawback, an alternative Euler angles error can be considered which is based on the rotation matrix describing the mutual orientation between Σ_d and Σ_e , i.e. ${}^e\mathbf{R}_d = \mathbf{R}_e^T \mathbf{R}_d$. Let $\boldsymbol{\varphi}_{de}$ denote the set of Euler angles that can be extracted from ${}^e\mathbf{R}_d$. In this case, the resolved angular acceleration can be chosen as

$$\mathbf{a}_o = \dot{\boldsymbol{\omega}}_d + \mathbf{T}_e(\boldsymbol{\varphi}_{de})(\mathbf{K}_{Do} \dot{\boldsymbol{\varphi}}_{de} + \mathbf{K}_{Po} \boldsymbol{\varphi}_{de}) - \dot{\mathbf{T}}_e(\boldsymbol{\varphi}_{de}, \dot{\boldsymbol{\varphi}}_{de}) \dot{\boldsymbol{\varphi}}_{de} \quad (2.12)$$

where $\mathbf{T}_e = \mathbf{R}_e \mathbf{T}$. The control law (2.12) ensures exponential tracking of the desired orientation trajectory for any choice of the positive definite matrix gains \mathbf{K}_{Do} and \mathbf{K}_{Po} , provided that $\mathbf{T}(\boldsymbol{\varphi}_{de})$ is nonsingular. The advantage of the alternative over the classical Euler angles error is that, by adopting a representation $\boldsymbol{\varphi}_{de}$ for which $\mathbf{T}(\mathbf{0})$ is nonsingular (e.g. the XYZ Euler angles), representation singularities occur only for large orientation errors.

A different definition of orientation error can be obtained using an angle/axis representation. The mutual orientation between Σ_d and Σ_e is described by ${}^e\mathbf{R}_d$, and thus the orientation error can be defined in terms of the general expression

$${}^e\mathbf{o}_{de} = f(\vartheta_{de}) {}^e\mathbf{r}_{de}, \quad (2.13)$$

where ϑ_{de} and ${}^e\mathbf{r}_{de}$ are respectively the rotation and the unit vector corresponding to ${}^e\mathbf{R}_d$, and $f(\cdot)$ is a suitable scalar function with $f(0) = 0$. Common choices for $f(\vartheta)$ are reported in Tab. 2.1.

<i>Representation</i>	<i>f(ϑ)</i>
Classical angle/axis	$\sin(\vartheta)$
Quaternion	$\sin(\vartheta/2)$
Rodrigues parameters	$\tan(\vartheta/2)$
Simple rotation	ϑ

Table 2.1. Common choices for $f(\vartheta)$

A special type of angle/axis representation of the orientation error is obtained with the quaternion, i.e.

$${}^e\mathbf{o}_{de} = \sin \frac{\vartheta_{de}}{2} {}^e\mathbf{r}_{de} = {}^e\boldsymbol{\epsilon}_{de} \quad (2.14)$$

corresponding to the vector part of the quaternion $\mathcal{Q}_{de} = \{\eta_{de}, {}^e\boldsymbol{\epsilon}_{de}\}$ that can be extracted from the rotation matrix ${}^e\mathbf{R}_d$. A brief review of the basic properties of the unit quaternion can be found, e.g. in [25].

The resolved angular acceleration based on the quaternion error can be chosen as

$$\mathbf{a}_o = \dot{\boldsymbol{\omega}}_d + \mathbf{K}_{Do} \Delta \boldsymbol{\omega}_{de} + \mathbf{K}_{Po} \mathbf{R}_e {}^e\boldsymbol{\epsilon}_{de} \quad (2.15)$$

where $\Delta \boldsymbol{\omega}_{de} = \boldsymbol{\omega}_d - \boldsymbol{\omega}_e$ is the relative angular velocity between Σ_d and Σ_e . Control law (2.15) ensures asymptotic tracking of the desired trajectory when the feedback gains are taken as scalar matrices, i.e. $\mathbf{K}_{Do} = k_{Do} \mathbf{I}$ and $\mathbf{K}_{Po} = k_{Po} \mathbf{I}$ [6].

3 Six-DOF Impedance Control

When the manipulator moves in free space, the end effector is required to match a desired frame Σ_d . Instead, when the end effector interacts with the environment, it is worth considering another (compliant) frame Σ_c specified by \mathbf{p}_c and \mathbf{R}_c ; then, a mechanical impedance can be introduced which is aimed at imposing a dynamic behaviour for the position and orientation displacements between the above two frames.

In the following, six-DOF impedance control schemes are derived using different types of orientation displacements.

3.1 Classical Impedance Control

The mutual position between the compliant frame and the desired frame can be described by the position displacement $\Delta \mathbf{p}_{dc} = \mathbf{p}_d - \mathbf{p}_c$. The impedance equation is typically chosen so as to enforce an equivalent mass-damper-spring behaviour for the end-effector position displacement under an external force \mathbf{f} acting on the end effector

$$\mathbf{M}_p \Delta \ddot{\mathbf{p}}_{dc} + \mathbf{D}_p \Delta \dot{\mathbf{p}}_{dc} + \mathbf{K}_p \Delta \mathbf{p}_{dc} = \mathbf{f}, \quad (3.1)$$

where \mathbf{M}_p , \mathbf{D}_p and \mathbf{K}_p are positive definite matrices.

In order to ensure a proper end-effector behaviour for the successful execution of an interaction task, the selection of the stiffness matrix plays a key role. Therefore, it is worth analysing the elastic term from a geometric point of view. The stiffness matrix \mathbf{K}_p can be decomposed as $\mathbf{K}_p = \mathbf{U}_p \mathbf{\Gamma}_p \mathbf{U}_p^T$ where $\mathbf{\Gamma}_p = \text{diag}\{\gamma_{p1}, \gamma_{p2}, \gamma_{p3}\}$ and $\mathbf{U}_p = [\mathbf{u}_{p1} \ \mathbf{u}_{p2} \ \mathbf{u}_{p3}]$ are respectively the eigenvalue matrix and the (orthogonal) eigenvector matrix. Then, considering a position displacement of length λ along the i -th eigenvector \mathbf{u}_{pi} leads to an elastic force

$$\mathbf{f}_E = \mathbf{K}_p \Delta \mathbf{p}_{dc} = \gamma_{pi} \lambda \mathbf{u}_{pi} \quad (3.2)$$

which is aligned with the same axis. This implies that the translational stiffness matrix can be expressed in terms of three parameters γ_{pi} representing the stiffness along three principal axes \mathbf{u}_{pi} , and in turn it establishes the property of task geometric consistency for the elastic force in (3.2).

The end-effector orientation displacement can be computed as a difference of Euler angles, i.e. $\Delta \phi_{dc} = \phi_d - \phi_c$, where ϕ_c and ϕ_d denote the set of Euler angles corresponding to \mathbf{R}_c and \mathbf{R}_d , respectively. In this case the rotational part of the impedance at the end effector can be formally defined in the same way as for the positional part (3.1), i.e.

$$\mathbf{M}_o \Delta \ddot{\boldsymbol{\varphi}}_{dc} + \mathbf{D}_o \Delta \dot{\boldsymbol{\varphi}}_{dc} + \mathbf{K}_o \Delta \boldsymbol{\varphi}_{dc} = \mathbf{T}^T(\boldsymbol{\varphi}_c) \boldsymbol{\mu}, \quad (3.3)$$

where \mathbf{M}_o , \mathbf{D}_o , \mathbf{K}_o are positive definite matrices describing the generalised inertia, rotational damping, rotational stiffness, respectively, and $\boldsymbol{\mu}$ is the contact moment at the end effector; all the above quantities have been referred to the base frame. Notice that, differently from (3.1), the dynamic behaviour for the rotational part is not absolutely determined by the choice of the impedance parameters but it does also depend on the orientation of the compliant frame with respect to the base frame through the matrix $\mathbf{T}^T(\boldsymbol{\varphi}_c)$. Moreover, Equation (3.3) becomes ill-defined in the neighborhood of a representation singularity; in particular, at such a singularity, moment components in the null space of \mathbf{T}^T do not generate any contribution to the dynamics of the orientation displacement, leading to a possible build-up of large values of contact moment. The effect of the rotational stiffness can be better understood by considering an infinitesimal orientation displacement between Σ_d

and Σ_c . From (3.3), in the absence of representation singularities, the elastic moment is $\boldsymbol{\mu}_E = \mathbf{T}^{-T}(\boldsymbol{\varphi}_c) \mathbf{K}_o \Delta \boldsymbol{\varphi}_{dc}$. In the case of an infinitesimal orientation displacement about $\boldsymbol{\varphi}_c$, it is

$$d(\Delta \boldsymbol{\varphi}_{dc}) = (\dot{\boldsymbol{\varphi}}_d - \dot{\boldsymbol{\varphi}}_c) \Big|_{\boldsymbol{\varphi}_d = \boldsymbol{\varphi}_c} dt = \mathbf{T}^{-1}(\boldsymbol{\varphi}_c) \Delta \boldsymbol{\omega}_{dc} dt, \quad (3.4)$$

where $\Delta \boldsymbol{\omega}_{dc} = \boldsymbol{\omega}_d - \boldsymbol{\omega}_c$ is the relative angular velocity between Σ_d and Σ_c . Hence, the elastic moment for an infinitesimal displacement $d(\Delta \boldsymbol{\varphi}_{dc})$ is

$$\boldsymbol{\mu}_E = \mathbf{T}^{-T}(\boldsymbol{\varphi}_c) \mathbf{K}_o \mathbf{T}^{-1}(\boldsymbol{\varphi}_c) \Delta \boldsymbol{\omega}_{dc} dt. \quad (3.5)$$

Equation (3.5) reveals that the relationship between the orientation displacement and the elastic moment depends on the orientation of Σ_c . It follows that the property of task geometric consistency of the elastic force (3.2) is lost, that is, the eigenvectors of the matrix \mathbf{K}_o do not represent the three principal axes for the rotational stiffness.

The drawbacks discussed above can be mitigated by adopting the alternative Euler angles displacement $\boldsymbol{\varphi}_{dc}$ that can be extracted from the rotation matrix ${}^c\mathbf{R}_d = \mathbf{R}_c^T \mathbf{R}_d$. Then, a rotational impedance at the end effector can be defined as

$$\mathbf{M}_o \ddot{\boldsymbol{\varphi}}_{dc} + \mathbf{D}_o \dot{\boldsymbol{\varphi}}_{dc} + \mathbf{K}_o \boldsymbol{\varphi}_{dc} = \mathbf{T}^T(\boldsymbol{\varphi}_{dc}) {}^c\boldsymbol{\mu} \quad (3.6)$$

where \mathbf{M}_o , \mathbf{D}_o and \mathbf{K}_o are defined in a similar way to (3.3) and ${}^c\boldsymbol{\mu}$ is referred to Σ_c . An advantage with respect to (3.3) is that now the impedance behaviour for the rotational part depends only on the relative orientation between Σ_d and Σ_c through the matrix $\mathbf{T}^T(\boldsymbol{\varphi}_{dc})$. Hence, if XYZ Euler angles are adopted, representation singularities have a mitigated effect since they occur for large end-effector orientation displacements. From (3.6) the elastic moment is ${}^c\boldsymbol{\mu}_E = \mathbf{T}^{-T}(\boldsymbol{\varphi}_{dc}) \mathbf{K}_o \boldsymbol{\varphi}_{dc}$. The infinitesimal orientation displacement about $\boldsymbol{\varphi}_{dc} = \mathbf{0}$ is

$$d\boldsymbol{\varphi}_{dc} = \dot{\boldsymbol{\varphi}}_{dc} \Big|_{\boldsymbol{\varphi}_{dc} = \mathbf{0}} dt = \mathbf{T}^{-1}(\mathbf{0}) \Delta^c \boldsymbol{\omega}_{dc} dt. \quad (3.7)$$

Hence, the elastic moment for an infinitesimal displacement $d\boldsymbol{\varphi}_{dc}$ is

$${}^c\boldsymbol{\mu}_E = \mathbf{T}^{-T}(d\boldsymbol{\varphi}_{dc}) \mathbf{K}_o \mathbf{T}^{-1}(\mathbf{0}) \Delta^c \boldsymbol{\omega}_{dc} dt = \mathbf{T}^{-T}(\mathbf{0}) \mathbf{K}_o \mathbf{T}^{-1}(\mathbf{0}) \Delta^c \boldsymbol{\omega}_{dc} dt \quad (3.8)$$

where the first-order approximation $\mathbf{T}^{-T}(d\boldsymbol{\varphi}_{dc}) dt \simeq \mathbf{T}^{-T}(\mathbf{0}) dt$ has been made. Equation (3.8) reveals that the relationship between the orientation displacement and the elastic moment is independent of the orientation of Σ_c . Notice, however, that the choice of Euler angles affects the resulting stiffness through the matrix $\mathbf{T}(\mathbf{0})$ which must be invertible. It is convenient to adopt the XYZ representation, which gives $\mathbf{T}(\mathbf{0}) = \mathbf{I}$ and thus, for an infinitesimal displacement,

$${}^c\boldsymbol{\mu}_E \simeq \mathbf{K}_o \Delta^c \boldsymbol{\omega}_{dc} dt. \quad (3.9)$$

As regards the property of task geometric consistency for the elastic moment, when \mathbf{K}_o is a diagonal matrix and the XYZ representation of Euler angles is adopted, the i -th eigenvector \mathbf{u}_{oi} of $\mathbf{K}_o = \text{diag}\{\gamma_{o1}, \gamma_{o2}, \gamma_{o3}\}$ is the i -th column of the identity matrix. Hence, the orientation displacement of an angle ϑ_{dc} about \mathbf{u}_{oi} is described by $\boldsymbol{\varphi}_{dc} = \vartheta_{dc}\mathbf{u}_{oi}$, which, in view of the expression of $\mathbf{T}(\boldsymbol{\varphi}_{dc})$ for XYZ Euler angles [25], leads to

$${}^c\boldsymbol{\mu}_E = \gamma_{oi}\vartheta_{dc}\mathbf{u}_{oi}, \quad (3.10)$$

representing an elastic moment about the same \mathbf{u}_{oi} axis; thus the vectors \mathbf{u}_{oi} have the meaning of rotational stiffness principal axes. It can be recognised that the same property does not hold in general for a nondiagonal \mathbf{K}_o [7].

3.2 Geometrically Consistent Impedance Control

The above analysis has revealed that the adoption of minimal representations of orientation for the rotational part of the impedance does not preserve properties analogous to those of the translational part. The main objective of this section is to define a six-DOF impedance with a rotational part matching the following desirable properties:

- the velocity used in the impedance equation should be dual to the moment $\boldsymbol{\mu}$ acting on the end effector, i.e. with no need of a transformation matrix depending on the actual end-effector orientation;
- the equivalent rotational stiffness in case of small orientation displacements should be always well defined;
- the elastic contribution should allow the specification of a rotational stiffness matrix in a consistent way with the task geometry.

A class of geometrically meaningful representations of the mutual orientation between Σ_d and Σ_c can be given in terms of the angle/axis displacement

$${}^c\mathbf{o}_{dc} = f(\vartheta_{dc}){}^c\mathbf{r}_{dc}, \quad (3.11)$$

where ϑ_{dc} and ${}^c\mathbf{r}_{dc}$ correspond to ${}^c\mathbf{R}_d$, and $f(\vartheta_{dc})$ is any of the functions listed in Tab. 2.1. Those are strictly increasing smooth functions in an interval $(-\vartheta_M, \vartheta_M)$ with $\vartheta_M > 0$. Hence, the derivative $f'(\vartheta_{dc})$ of f with respect to ϑ_{dc} is strictly positive in that interval.

Differentiating (3.11), gives [7]

$${}^c\dot{\mathbf{o}}_{dc} = \boldsymbol{\Omega}({}^c\mathbf{r}_{dc}, \vartheta_{dc})\Delta^c\boldsymbol{\omega}_{dc} \quad (3.12)$$

with

$$\boldsymbol{\Omega} = f'(\vartheta_{dc}){}^c\mathbf{r}_{dc}{}^c\mathbf{r}_{dc}^T + \frac{1}{2}f(\vartheta_{dc})\left(\cot(\vartheta_{dc}/2)(\mathbf{I} - {}^c\mathbf{r}_{dc}{}^c\mathbf{r}_{dc}^T) - \mathbf{S}({}^c\mathbf{r}_{dc})\right), \quad (3.13)$$

where $\mathbf{S}(\cdot)$ is the (3×3) skew-symmetric matrix operator performing the cross product. Notice that the following property of $\boldsymbol{\Omega}$ holds

$$\boldsymbol{\Omega}({}^c\mathbf{r}_{dc}, 0) = f'(0)\mathbf{I} \quad (3.14)$$

which will be useful in the following.

In order to derive the impedance equation for the rotational part, it is convenient to refer to the following energy-based argument. Let

$$\mathcal{T}_o = \frac{1}{2} \Delta^c \boldsymbol{\omega}_{dc}^T \mathbf{M}_o \Delta^c \boldsymbol{\omega}_{dc} \quad (3.15)$$

express a rotational pseudo-kinetic energy of a rigid body with inertia tensor \mathbf{M}_o and angular velocity $\Delta^c \boldsymbol{\omega}_{dc}$. Then consider the potential energy

$$\mathcal{U}_o = \psi^c \mathbf{o}_{dc}^T \mathbf{K}_o^c \mathbf{o}_{dc}, \quad (3.16)$$

where ψ is a positive constant depending on the particular choice of $f(\cdot)$, and \mathbf{K}_o is a symmetric positive definite matrix. Having defined the various energy contributions, the terms in the rotational impedance equation can be derived by considering the associated powers. Taking the time derivative of (3.15) yields

$$\dot{\mathcal{T}}_o = {}^c\boldsymbol{\mu}_I^T \Delta^c \boldsymbol{\omega}_{dc} \quad (3.17)$$

where

$${}^c\boldsymbol{\mu}_I = \mathbf{M}_o \Delta^c \dot{\boldsymbol{\omega}}_{dc} \quad (3.18)$$

is the inertial moment. Further, taking the time derivative of (3.16) and accounting for (3.12) yields

$$\dot{\mathcal{U}}_o = {}^c\boldsymbol{\mu}_E^T \Delta^c \boldsymbol{\omega}_{dc} \quad (3.19)$$

where

$${}^c\boldsymbol{\mu}_E = 2\psi \boldsymbol{\Omega}^T({}^c\mathbf{r}_{dc}, \vartheta_{dc}) \mathbf{K}_o^c \mathbf{o}_{dc} \quad (3.20)$$

is the elastic moment. Finally, a dissipative contribution can be added as

$${}^c\boldsymbol{\mu}_D = \mathbf{D}_o \Delta^c \boldsymbol{\omega}_{dc}, \quad (3.21)$$

where \mathbf{D}_o is a positive definite matrix characterising a rotational damping at the end effector.

Therefore, a rotational impedance at the end effector can be defined by adding the contributions (3.18), (3.21) and (3.20), i.e.

$$\mathbf{M}_o \Delta^c \dot{\boldsymbol{\omega}}_{dc} + \mathbf{D}_o \Delta^c \boldsymbol{\omega}_{dc} + \mathbf{K}_o' {}^c \mathbf{o}_{dc} = {}^c\boldsymbol{\mu}, \quad (3.22)$$

where the equality ${}^c\boldsymbol{\mu} = {}^c\boldsymbol{\mu}_I + {}^c\boldsymbol{\mu}_D + {}^c\boldsymbol{\mu}_E$ has been imposed, and

$$\mathbf{K}_o' = 2\psi \boldsymbol{\Omega}^T({}^c\mathbf{r}_{dc}, \vartheta_{dc}) \mathbf{K}_o. \quad (3.23)$$

Notice that the rotational part of the impedance equation has been derived in terms of quantities all referred to Σ_c ; this allows the impedance

behaviour to be effectively expressed in terms of the relative orientation between Σ_d and Σ_c , no matter what the absolute orientation of the compliant frame with respect to the base frame is. It is worth remarking that the velocity used in the impedance equation is dual to the moment ${}^c\boldsymbol{\mu}$ exerted by the end effector, i.e. there is no need of a transformation matrix depending on the actual end-effector orientation.

In the following, the analysis for small orientation displacements is carried out and consistency with the task geometry is investigated. Consider an infinitesimal orientation displacement expressed as

$$d^c\mathbf{o}_{dc} = {}^c\dot{\mathbf{o}}_{dc} \Big|_{\vartheta_{dc}=0} dt = \boldsymbol{\Omega}({}^c\mathbf{r}_{dc}, 0) \Delta^c \boldsymbol{\omega}_{dc} dt = f'(0) \Delta^c \boldsymbol{\omega}_{dc} dt \quad (3.24)$$

where the property (3.14) has been exploited. Folding (3.24) into (3.20), written for an infinitesimal displacement about $\vartheta_{dc} = 0$, gives

$${}^c\boldsymbol{\mu}_E = 2\psi \boldsymbol{\Omega}^T({}^c\mathbf{r}_{dc}, d\vartheta_{dc}) \mathbf{K}_o d^c\mathbf{o}_{dc} \quad (3.25)$$

$$\simeq 2\psi (f'(0))^2 \mathbf{K}_o \Delta^c \boldsymbol{\omega}_{dc} dt = \mathbf{K}_o \Delta^c \boldsymbol{\omega}_{dc} dt, \quad (3.26)$$

where the first-order approximation $\boldsymbol{\Omega}({}^c\mathbf{r}_{dc}, d\vartheta_{dc}) \simeq f'(0) \mathbf{I}$ has been considered and the choice $\psi = 1/2(f'(0))^2$ has been made. Equation (3.25) clearly shows how the relationship between the orientation displacement and the elastic moment is independent of the orientation of Σ_c , and the problem of representation singularities is not of concern since $f'(0)$ is finite.

As regards the property of task geometric consistency, the stiffness matrix in (3.20) can be decomposed as $\mathbf{K}_o = \mathbf{U}_o \boldsymbol{\Gamma}_o \mathbf{U}_o^T$, where $\boldsymbol{\Gamma}_o = \text{diag}\{\gamma_{o1}, \gamma_{o2}, \gamma_{o3}\}$ and $\mathbf{U}_o = [\mathbf{u}_{o1} \ \mathbf{u}_{o2} \ \mathbf{u}_{o3}]$ are respectively the eigenvalue matrix and the (orthogonal) eigenvector matrix. Then, considering an orientation displacement by an angle ϑ_{dc} about the i -th eigenvector

$${}^c\mathbf{o}_{dc} = f(\vartheta_{dc}) \mathbf{u}_{oi}, \quad (3.27)$$

and taking into account the expression of $\boldsymbol{\Omega}$ in (3.13), yields

$${}^c\boldsymbol{\mu}_E = 2\psi f'(\vartheta_{dc}) f(\vartheta_{dc}) \gamma_{oi} \mathbf{u}_{oi}. \quad (3.28)$$

This represents an elastic moment about the same \mathbf{u}_{oi} axis which is in the same direction of the orientation displacement since $f'(\vartheta_{dc}) > 0$. Therefore, the rotational stiffness matrix can be expressed in terms of three parameters γ_{oi} representing the stiffness about three principal axes \mathbf{u}_{oi} , i.e. in a consistent way with the task geometry.

A special case among angle/axis representations of orientation displacement is constituted by the quaternion displacement. Such a representation has the advantage over other angle/axis representations to avoid representation singularities. The mutual orientation between Σ_d and Σ_c can be described by the quaternion $\mathcal{Q}_{dc} = \{\eta_{dc}, {}^c\boldsymbol{\epsilon}_{dc}\}$ extracted from ${}^c\mathbf{R}_d$. Indeed, the orientation displacement to be considered in (3.22) is given by the vector part ${}^c\boldsymbol{\epsilon}_{dc}$ and, in view of (3.16), the expression of the potential energy becomes

$$\mathcal{U}_o = 2 {}^c\epsilon_{dc}^T \mathbf{K}_o {}^c\epsilon_{dc}, \quad (3.29)$$

where it has been set $\psi = 2$. Even though the potential energy is expressed in terms of the vector part of the quaternion, it can be shown that \mathcal{U}_o coincides with the rotational elastic energy associated with a torsional spring of stiffness \mathbf{K}_o acting so as to align Σ_c with Σ_d [7].

In view of (3.22), the resulting impedance equation for the rotational part becomes

$$\mathbf{M}_o \Delta^c \dot{\boldsymbol{\omega}}_{dc} + \mathbf{D}_o \Delta^c \boldsymbol{\omega}_{dc} + \mathbf{K}'_o {}^c\epsilon_{dc} = {}^c\boldsymbol{\mu}, \quad (3.30)$$

where the rotational stiffness matrix is

$$\mathbf{K}'_o = 2\mathbf{E}^T(\eta_{dc}, {}^c\epsilon_{dc})\mathbf{K}_o \quad (3.31)$$

where $\mathbf{E} = \eta_{dc}\mathbf{I} - \mathbf{S}({}^c\epsilon_{dc})$.

A generalisation of the proposed approach leads to an impedance equation characterised by coupling elastic terms between the translational and the rotational part; they may be useful for the execution of assembly tasks. The details can be found in [8].

3.3 Inner Motion Control Loop

The impedance behaviour at the end-effector can be enforced by resorting to a model-based control law designed so as to give closed-loop dynamic equations, for both position and orientation, coinciding with those describing the desired impedance [24]. However, the selection of good impedance parameters that guarantee a satisfactory compliant behaviour during the interaction may turn out to be inadequate to ensure accurate tracking of the desired position and orientation trajectory when the end effector moves in free space. A solution to this drawback can be devised by separating the motion control action from the impedance control action as follows. The motion control action is purposefully made stiff so as to enhance disturbance rejection but, rather than ensuring tracking of the desired end-effector position and orientation, it shall ensure tracking of a reference position and orientation resulting from the impedance control action [1]. In other words, the desired position and orientation together with the measured contact force and moment are input to the impedance equation which, via a suitable integration, generates the position and orientation to be used as a reference for the motion control action.

In detail, the control input \mathbf{a}_p in (2.6) has to be designed to match the desired impedance for the translational part and the rotational part, respectively. In view of (3.1), \mathbf{a}_p is taken as in (2.9), by simply replacing the variables referring to the desired frame (subscript d) with the variables referring to the compliant frame (subscript c). The latter can be computed by forward integration of the translational impedance equation (3.1) with input \mathbf{f} available from the force sensor.

As regards the orientation loop, \mathbf{a}_o can be chosen according to the different representations of orientation displacement illustrated above. Similarly to what has been conceived for \mathbf{a}_p , the control input \mathbf{a}_o is taken as in (2.10), (2.12) or (2.15) by simply replacing the variables referring to the desired frame (subscript d) with the variables referring to the compliant frame (subscript c). The latter can be computed by forward integration of the corresponding rotational impedance equations (3.3), (3.6) or (3.30) with input $\boldsymbol{\mu}$ available from the force sensor.

3.4 Redundancy Resolution

In case the manipulator is kinematically redundant, there exist infinite joint motions that produce the same end-effector motion. In particular, even when the end effector is at rest, it is possible to generate an internal motion at the joints. As a minimal requirement, such motion should be made stable [16]. In addition, it could be exploited to meet additional task requirements besides the execution of the end-effector trajectory, thus providing redundancy resolution [19].

Redundancy can be solved either at kinematic level, that is in the first stage of a kinematic control strategy, or at dynamic level by suitably modifying the inverse dynamics control law [13]. The latter approach is pursued hereafter.

Since the Jacobian matrix for a redundant manipulator has more columns than rows ($n > 6$), a suitable right inverse of \mathbf{J} is to be used in lieu of \mathbf{J}^{-1} . Hence, in lieu of (2.5), the new control input in (2.3) can be chosen as

$$\boldsymbol{\alpha} = \mathbf{J}^\dagger(\mathbf{q}) \left(\mathbf{a} - \dot{\mathbf{J}}(\mathbf{q})\dot{\mathbf{q}} \right) + \boldsymbol{\alpha}_n \quad (3.32)$$

where

$$\mathbf{J}^\dagger = \mathbf{W}^{-1} \mathbf{J}^T \left(\mathbf{J} \mathbf{W}^{-1} \mathbf{J}^T \right)^{-1} \quad (3.33)$$

denotes the right pseudo-inverse of \mathbf{J} weighted by the positive definite ($n \times n$) matrix \mathbf{W} . Also, in (3.32), $\boldsymbol{\alpha}_n$ denotes a joint acceleration vector lying in the null space of \mathbf{J} which is available for redundancy resolution. It can be shown that plugging (3.32) in (2.4) yields the same end-effector resolved acceleration as in (2.6) [20].

The matrix projecting arbitrary joint accelerations into the null space of \mathbf{J} is given by $(\mathbf{I} - \mathbf{J}^\dagger \mathbf{J})$, no matter what choice is made for the weighting matrix \mathbf{W} in (3.33). Therefore, it is significant to choose \mathbf{W} so that the redundancy resolution scheme for motion control should not be altered when interaction with the environment occurs. To this purpose, from (2.2) the joint accelerations induced by the external end-effector force and moment are given by

$$\ddot{\mathbf{q}}_e = -\mathbf{B}^{-1}(\mathbf{q}) \mathbf{J}^T(\mathbf{q}) \mathbf{h}. \quad (3.34)$$

Projecting these accelerations in the null space of the Jacobian gives

$$\ddot{\mathbf{q}}_{en} = - \left(\mathbf{I} - \mathbf{J}^\dagger(\mathbf{q})\mathbf{J}(\mathbf{q}) \right) \mathbf{B}^{-1}(\mathbf{q})\mathbf{J}^T(\mathbf{q})\mathbf{h}. \quad (3.35)$$

Choosing $\mathbf{W} = \mathbf{B}$ in (3.33) and plugging the resulting \mathbf{J}^\dagger in (3.35) yields $\ddot{\mathbf{q}}_{en} = \mathbf{0}$, meaning that the external force and moment produce no null space joint accelerations. Therefore, in view of this choice, in the design of the joint resolved acceleration in (3.32) the vector $\boldsymbol{\alpha}_n$ can be used to solve redundancy independently of the occurrence of interaction with the environment. The matrix

$$\mathbf{J}^\dagger = \mathbf{B}^{-1}\mathbf{J}^T \left(\mathbf{J}\mathbf{B}^{-1}\mathbf{J}^T \right)^{-1} \quad (3.36)$$

weighted by the inertia matrix is termed dynamically consistent pseudo-inverse of the Jacobian.

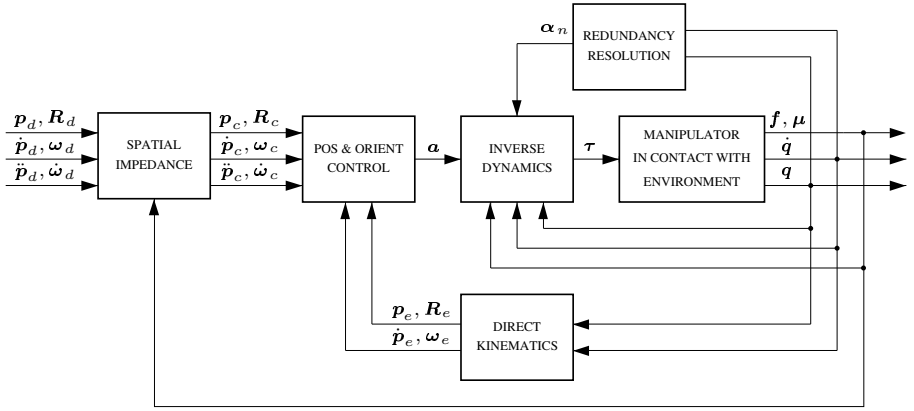


Figure 3.1. Spatial impedance control with redundancy resolution

The next step consists of designing a redundancy resolution control in terms of the null space joint accelerations $\boldsymbol{\alpha}_n$ in (3.32). To this purpose, $\boldsymbol{\alpha}_n$ shall be chosen so as to ensure stabilisation of the null space motion and possibly optimisation of an additional task function. Let

$$\mathbf{e}_n = \left(\mathbf{I} - \mathbf{J}^\dagger(\mathbf{q})\mathbf{J}(\mathbf{q}) \right) (\boldsymbol{\beta} - \dot{\mathbf{q}}) \quad (3.37)$$

denote the null space velocity error where $\boldsymbol{\beta}$ is a joint velocity vector which is available for redundancy resolution. The goal is to make \mathbf{e}_n asymptotically converge to zero. Choosing

$$\boldsymbol{\alpha}_n = (\mathbf{I} - \mathbf{J}^\dagger\mathbf{J}) \left(\dot{\boldsymbol{\beta}} - \dot{\mathbf{J}}^\dagger\mathbf{J}(\boldsymbol{\beta} - \dot{\mathbf{q}}) + \mathbf{B}^{-1}(\mathbf{K}_n\mathbf{e}_n + \mathbf{C}\mathbf{e}_n) \right) \quad (3.38)$$

where \mathbf{K}_n is a positive definite matrix, it can be shown that $\mathbf{e}_n \rightarrow \mathbf{0}$ asymptotically [20].

Regarding the utilisation of redundancy, a typical choice for β is

$$\beta = k_\beta \mathbf{B}^{-1} \left(\frac{\partial w(\mathbf{q})}{\partial \mathbf{q}} \right) \quad (3.39)$$

where k_β is a signed scalar and $w(\mathbf{q})$ is an additional task function that can be locally optimised.

A block diagram summarising the overall spatial impedance control with redundancy resolution is sketched in Figure 3.1.

4 Cooperative Robots

4.1 Loose Cooperation

Consider a system of two robots manipulating an object. A cooperative control strategy can be termed loose when the manipulation task is executed by controlling the two robots in an independent fashion. Cooperation is realised only at the task planning level.

A typical task requiring loose cooperative control is constituted by mating rigid parts such as dual-robot assembly in a workcell. The archetype is the classical peg-in-hole, where one robot carries the peg and the other holds the hollow part. It should be clear that the task is successfully executed provided that mating forces are suitably reduced during the insertion so as to avoid undesirable jamming and wedging. This concept can be brought to fruition by resorting to special mechanical devices such as the Remote Center of Compliance in [28] or the compliant end effectors in [11], [15].

An alternative strategy [5] is to assign complementary roles to the two robots, i.e. to operate one robot using pure positional control while controlling the other so as to achieve a programmable impedance at the end effector. In detail, the motion of the position controlled robot is planned to match the nominal requirements of the assigned task, while the active compliant behaviour imposed to the impedance controlled robot is devoted to mitigating the effects of imperfect knowledge of the task geometry and unavoidable tracking errors.

The position controlled robot can be operated using the standard industrial control unit, i.e. by exploiting the set of motion planning instructions of the native programming language.

On the other hand, assuming that an open control architecture is available for the other robot, an impedance control is realised at the end-effector level where the object is either the peg or the hollow part; the end-effector frame of such a robot coincides with the object frame.

4.2 Tight Cooperation

A cooperative control strategy can be termed tight when the manipulation task is executed by controlling the robots in a coordinated fashion. Cooperation is realised not only at the task planning level, but also at the control level. This is the typical task of two robots whose end effectors tightly grasp a commonly held rigid object, thus creating a closed-kinematic chain.

Consider a two-robot system tightly grasping a rigid object in contact with the environment. Let Σ_o be the frame attached to the object; its origin and orientation with respect to the base frame are characterised by the (3×1) position vector \mathbf{p}_o and the (3×3) rotation matrix \mathbf{R}_o , respectively.

The object motion must be related to the motions of the end effectors of the robots. This can be done by resorting to the task-oriented formulation for coordinated motion of dual-robot systems developed in [9], [3].

For each manipulator ($k = 1, 2$) let Σ_k denote a frame attached to the end effector; its origin and orientation with respect to the base frame are characterised by the (3×1) position vector \mathbf{p}_k and the (3×3) rotation matrix \mathbf{R}_k , respectively. Then, $\mathcal{Q}_k = \{\eta_k, \epsilon_k\}$ represents the unit quaternion corresponding to \mathbf{R}_k . Let also $\mathbf{v}_k = [\dot{\mathbf{p}}_k^T \ \boldsymbol{\omega}_k^T]^T$ be the (6×1) end-effector (linear and angular) velocity vector.

The object position is chosen as

$$\mathbf{p}_o = \frac{1}{2}(\mathbf{p}_1 + \mathbf{p}_2), \quad (4.1)$$

while the rotation matrix defining the orientation of Σ_o is chosen as

$$\mathbf{R}_o = \mathbf{R}_1^1 \mathbf{R}({}^1\mathbf{r}_{21}, \vartheta_{21}/2), \quad (4.2)$$

where ${}^1\mathbf{r}_{21}$ and ϑ_{21} are respectively the unit vector and the angle that realise the rotation described by

$${}^1\mathbf{R}_2 = \mathbf{R}_1^T \mathbf{R}_2 \quad (4.3)$$

and ${}^1\mathbf{R}({}^1\mathbf{r}_{21}, \vartheta_{21}/2)$ is the rotation matrix corresponding to a rotation of $\vartheta_{21}/2$ about the axis ${}^1\mathbf{r}_{21}$. Then the absolute orientation can be expressed as

$$\mathcal{Q}_o = \mathcal{Q}_1 * \left\{ \cos \frac{\vartheta_{21}}{4}, \sin \frac{\vartheta_{21}}{4} {}^1\mathbf{r}_{21} \right\} \quad (4.4)$$

where \mathcal{Q}_o is the unit quaternion corresponding to \mathbf{R}_o and “*” denotes the quaternion product (see the appendix in [25]); the second factor on the right-hand side of (4.4) is the unit quaternion extracted from ${}^1\mathbf{R}({}^1\mathbf{r}_{21}, \vartheta_{21}/2)$.

From (4.1) and (4.2), the object linear velocity $\dot{\mathbf{p}}_o$ and angular velocity $\boldsymbol{\omega}_o$ can be expressed as

$$\mathbf{v}_o = \frac{1}{2}(\mathbf{v}_1 + \mathbf{v}_2) \quad (4.5)$$

where $\mathbf{v}_o = [\dot{\mathbf{p}}_o^T \ \boldsymbol{\omega}_o^T]^T$.

Let \mathbf{f}_k and $\boldsymbol{\mu}_k$ ($k = 1, 2$) respectively denote the (3×1) end-effector force and moment vectors for either manipulator. Then, according to the kinetostatics duality concept [26] applied to (4.5), the object force and moment can be expressed as

$$\mathbf{h}_o = \mathbf{h}_1 + \mathbf{h}_2 \quad (4.6)$$

where $\mathbf{h}_k = [\mathbf{f}_k^T \quad \boldsymbol{\mu}_k^T]^T$ and $\mathbf{h}_o = [\mathbf{f}_o^T \quad \boldsymbol{\mu}_o^T]^T$.

In order to fully describe a coordinated motion, the position and orientation of one manipulator relative to the other is also of concern. The mutual position between the two end effectors is the vector

$$\Delta \mathbf{p}_{21} = \mathbf{p}_2 - \mathbf{p}_1. \quad (4.7)$$

The mutual orientation between the two end effectors is defined with reference to Σ_1 in terms of the rotation matrix ${}^1\mathbf{R}_2$, and then in terms of the quaternion product

$$\mathcal{Q}_{21} = \mathcal{Q}_1^{-1} * \mathcal{Q}_2, \quad (4.8)$$

where $\mathcal{Q}_1^{-1} = \{\eta_1, -\boldsymbol{\epsilon}_1\}$ is the unit quaternion corresponding to \mathbf{R}_1^T (i.e. the conjugate of \mathcal{Q}_1).

From (4.3) and (4.7), the mutual velocity can be expressed as

$$\Delta \mathbf{v}_{21} = \mathbf{v}_2 - \mathbf{v}_1. \quad (4.9)$$

Once the position and orientation variables describing the cooperative system have been defined, reference end-effector position $\mathbf{p}_{r,k}$ and orientation $\mathcal{Q}_{r,k}$ variables, as well as reference end-effector velocities $\mathbf{v}_{r,k}$ ($k = 1, 2$), must be generated with a twofold objective; namely, realising an impedance behaviour at the object level, while assigning a mutual position and orientation between the two end effectors that is compatible with the object geometry.

The first objective can be fulfilled as follows. Let the desired object position \mathbf{p}_d and orientation \mathcal{Q}_d (extracted from the desired rotation matrix \mathbf{R}_d) be assigned with the associated linear and angular velocities and accelerations. Also, the object force and moment can be computed from (4.6) with the end-effector forces and moments available from the wrist force/torque sensors. Then, the translational and rotational impedance equations (3.1) and (3.30) are integrated, with input \mathbf{f}_o and ${}^c\boldsymbol{\mu}_o$, to compute $\dot{\mathbf{p}}_c$ and ${}^c\dot{\boldsymbol{\omega}}_c$, $\dot{\mathbf{p}}_c$ and ${}^c\boldsymbol{\omega}_c$, and then \mathbf{p}_c and \mathcal{Q}_c via the quaternion propagation.

The second objective can be fulfilled by assigning a reference mutual position $\Delta \mathbf{p}_{r,21}$ and orientation $\mathcal{Q}_{r,21}$. In particular, $\Delta \mathbf{p}_{r,21}$ and $\mathcal{Q}_{r,21}$ are taken as constant and equal to the initial values of $\Delta \mathbf{p}_{21}$ in (4.7) and \mathcal{Q}_{21} extracted from (4.3), respectively, that can be computed via the direct kinematics of the two manipulators.

The two objectives are combined by choosing the above reference position and orientation for the two end effectors so as to satisfy (4.1), (4.7), and (4.4), (4.8), i.e.

$$\mathbf{p}_{r,1} = \mathbf{p}_c - \frac{1}{2}\Delta\mathbf{p}_{r,21} \quad (4.10)$$

$$\mathbf{p}_{r,2} = \mathbf{p}_c + \frac{1}{2}\Delta\mathbf{p}_{r,21} \quad (4.11)$$

$$\mathbf{Q}_{r,1} = \mathbf{Q}_c * \left\{ \cos \frac{\vartheta_{r,21}}{4}, -\sin \frac{\vartheta_{r,21}}{4} \mathbf{r}_{r,21} \right\} \quad (4.12)$$

$$\mathbf{Q}_{r,2} = \mathbf{Q}_{r,1} * \mathbf{Q}_{r,21}. \quad (4.13)$$

Further, the reference velocities for the two end effectors are chosen as

$$\mathbf{v}_{r,1} = \mathbf{v}_c - \frac{1}{2}\Delta\mathbf{v}_{r,21} \quad (4.14)$$

$$\mathbf{v}_{r,2} = \mathbf{v}_c + \frac{1}{2}\Delta\mathbf{v}_{r,21} \quad (4.15)$$

where $\mathbf{v}_c = [\dot{\mathbf{p}}_c^T \ \omega_c^T]^T$. Then, the reference accelerations can be computed via a formal time derivative of the terms in (4.14) and (4.15).

The above reference trajectories can be tracked by resorting to an inner motion control loop based on an inverse dynamics strategy as for the case of the single manipulator.

5 Experimental Validation

The setup in the PRISMA Lab consists of two industrial robots Comau SMART-3 S (Figure 5.1). Each robot arm has a six-revolute-joint anthropomorphic geometry with nonnull shoulder and elbow offsets and non-spherical wrist. One arm is mounted on a sliding track which provides an additional degree of mobility. The joints are actuated by brushless motors via gear trains; shaft absolute resolvers provide motor position measurements. Each robot is controlled by the C3G 9000 control unit which has a VME-based architecture. Independent joint control is adopted where the individual servos are implemented as standard PID controllers. The native robot programming language is PDL 2, a high-level Pascal-like language with typical motion planning instructions.

An open version of the control unit is available [2] which allows testing of advanced control algorithms on a conventional industrial robot. Connection of the VME bus of the C3G 9000 unit to the ISA bus of a standard PC is made possible by a BIT 3 Computer bus adapter board; for the experiments, a PC Pentium MMX/233 was used.

Various operating modes are available in the control unit, allowing the PC to interact with the original controller both at trajectory generation level and at joint control level. To implement model-based control schemes, the operating mode number 4 is used in which the PC is in charge of computing the control algorithm and passing the references to the current servos

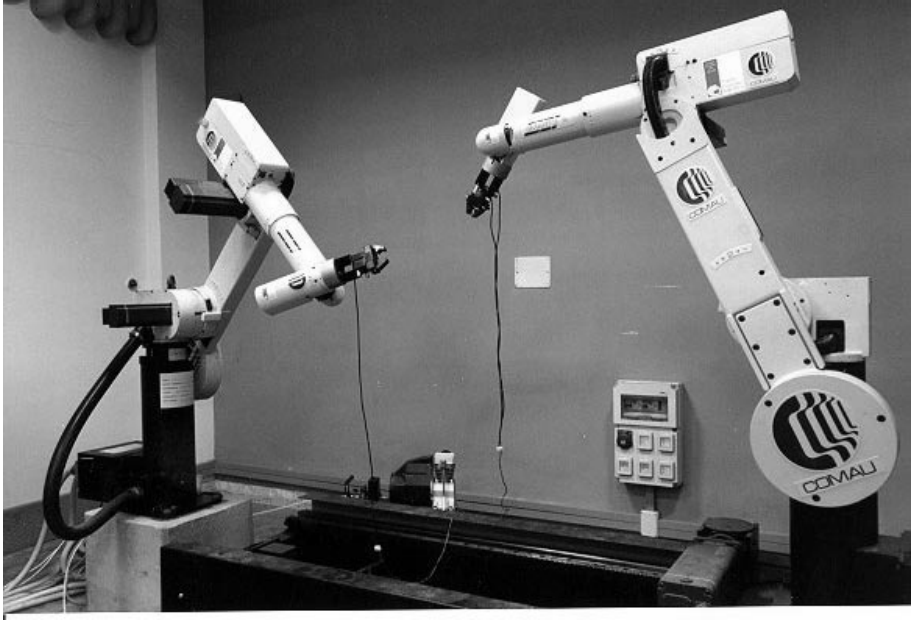


Figure 5.1. Experimental setup available in the PRISMA Lab

through the communication link at 1 ms sampling time. Joint velocities are reconstructed through numerical differentiation of joint position readings.

A six-axis force/torque sensor ATI FT30-100 with force range of ± 130 N and torque range of ± 10 N·m can be mounted at either arm's wrist. The sensor is connected to the PC by a parallel interface board which provides readings of six components of generalised force at 1 ms.

5.1 Single Manipulator

The above impedance control schemes have been tested in a number of experiments. An end effector has been built as a steel stick with a wooden disk of 5.5 cm radius at the tip. The end-effector frame has its origin at the center of the disk and its approach axis normal to the disk surface and pointing outwards.

First case study: Interaction with environment. The first case study has been developed to analyse interaction with environment. This is constituted by a flat plexiglas surface. The translational stiffness at the contact between the end effector and the surface is of the order of 10^4 N/m, while the rotational stiffness for small angles is of the order of 20 Nm/rad.

The task consists of taking the disk in contact with the surface at an angle of unknown magnitude (Figure 5.2). The end-effector desired position

is required to make a straight-line motion with a vertical displacement of -0.24 m along the Z_b -axis of the base frame. The trajectory along the path is generated according to a fifth-order interpolating polynomial with null initial and final velocities and accelerations, and a duration of 7 s. The end-effector desired orientation is required to remain constant during the task. The surface is placed (horizontally) in the X_bY_b -plane in such a way as to obstruct the desired end-effector motion, both for the translational part and for the rotational part.

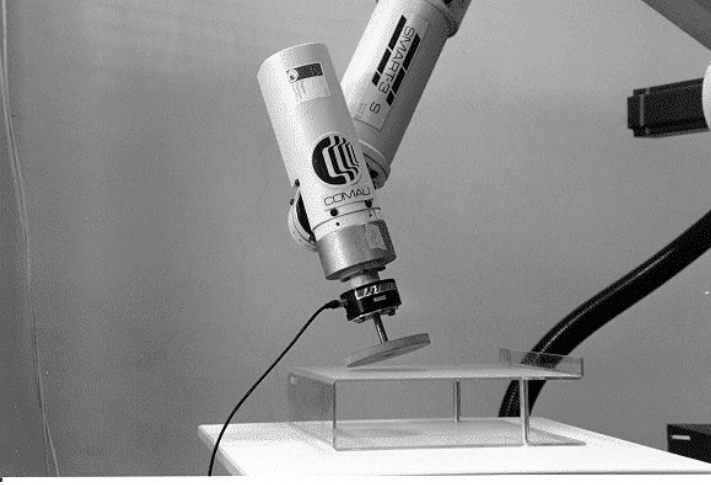


Figure 5.2. End effector in contact with plexiglas surface

The parameters of the translational part of the six-DOF impedance equation (3.1) have been set to $\mathbf{M}_p = 9\mathbf{I}$, $\mathbf{D}_p = 2000\mathbf{I}$, $\mathbf{K}_p = 700\mathbf{I}$, while the parameters of the rotational part of the six-DOF impedance equation (3.30) have been set to $\mathbf{M}_o = 0.4\mathbf{I}$, $\mathbf{D}_o = 5\mathbf{I}$, $\mathbf{K}_o = 2\mathbf{I}$. Notice that the stiffness matrices have been chosen so as to ensure a compliant behaviour at the end effector (limited values of contact force and moment) during the constrained motion, while the damping matrices have been chosen so as to guarantee a well-damped behaviour.

The gains of the inner motion control loop actions in (2.9),(2.15) have been set to $\mathbf{K}_{Pp} = 2025\mathbf{I}$, $\mathbf{K}_{Po} = 4500\mathbf{I}$, $\mathbf{K}_{Dp} = \mathbf{K}_{Do} = 65\mathbf{I}$.

The results in Figure 5.3 show the effectiveness of the quaternion-based six-DOF impedance control. After the contact, the component of the position error between Σ_d and Σ_e $\Delta \mathbf{p}_{de} = \mathbf{p}_d - \mathbf{p}_e$ along the Z_b -axis significantly deviates from zero, as expected, while small errors can be seen also for the components along the X_b - and the Y_b -axis due to contact friction. As for the orientation error, all the components of the orientation displacement be-

tween Σ_d and Σ_e (${}^e\epsilon_{de}$) significantly deviate from zero since the end-effector frame has to rotate with respect to the base frame after the contact in order to comply with the surface. Also, in view of the imposed task, a prevailing component of the contact force can be observed along the Z_b -axis after the contact, while the small components along the X_b - and the Y_b -axis arise as a consequence of the above end-effector deviation. As for the contact moment referred to Σ_d , the component about the Z_b -axis is small, as expected. It can be recognised that all the above quantities reach constant steady-state values after the desired motion is stopped. The oscillations on the force and moment during the transient can be mainly ascribed to slipping of the disk on the surface after the contact.

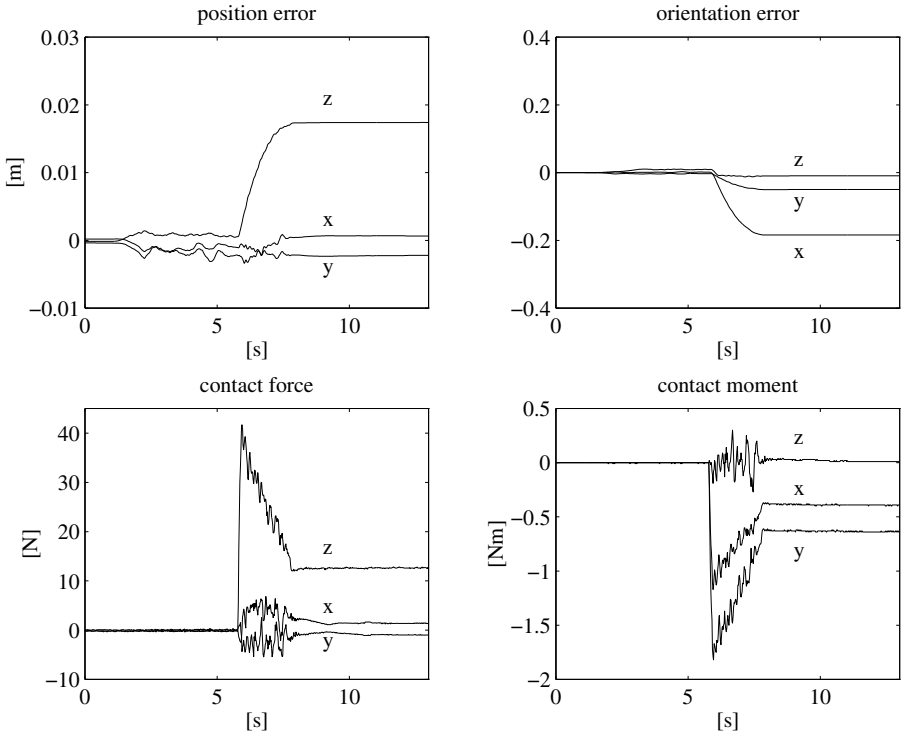


Figure 5.3. Experimental results under six-DOF impedance control based on quaternion in the first case study

In sum, it can be asserted that a compliant behaviour is successfully achieved. A similar performance has been obtained also with the six-DOF impedance control schemes based on the Euler angles error, i.e. by using either (3.3) or (3.6) in lieu of (3.30). This fact can be explained because both the absolute end-effector orientation in (3.3) and the relative orientation

in (3.6) keep far from representation singularities. The results are not reported here for brevity.

Second case study: Representation singularity. The second case study is aimed at testing the performance of the quaternion-based compared to the Euler angles-based six-DOF impedance control, when the end-effector orientation is close to a representation singularity of \mathbf{T} . The end effector and the surface are the same as in the previous case study.

The end-effector desired position is required to make a straight-line motion with a horizontal displacement of 0.085 m along the X_b -axis of the base frame. The trajectory along the path is generated according to a fifth-order interpolating polynomial with null initial and final velocities and accelerations, and a duration of 5 s. The end-effector desired orientation is required to remain constant during the task. The surface is now placed vertically in such a way as to obstruct the desired end-effector motion, only for the rotational part though. Therefore, no impedance control has been accomplished for the translational part, i.e. \mathbf{p}_c in (2.9) coincides with \mathbf{p}_d .

The parameters of the quaternion-based impedance equations (3.1),(3.30) are set to $\mathbf{M}_p = 10\mathbf{I}$, $\mathbf{D}_p = 600\mathbf{I}$, $\mathbf{K}_p = 1000\mathbf{I}$, $\mathbf{M}_o = 0.25\mathbf{I}$, $\mathbf{D}_o = 3.5\mathbf{I}$, $\mathbf{K}_o = 2.5\mathbf{I}$. In order to carry out a comparison, the impedance control based on the Euler angles has also been tested. The parameters of the rotational impedance equation (3.3) have been set to the same values as for the quaternion. As regards the gains of the inner motion control loop, these have been chosen equal to those in the previous experiment for both types of impedance control schemes.

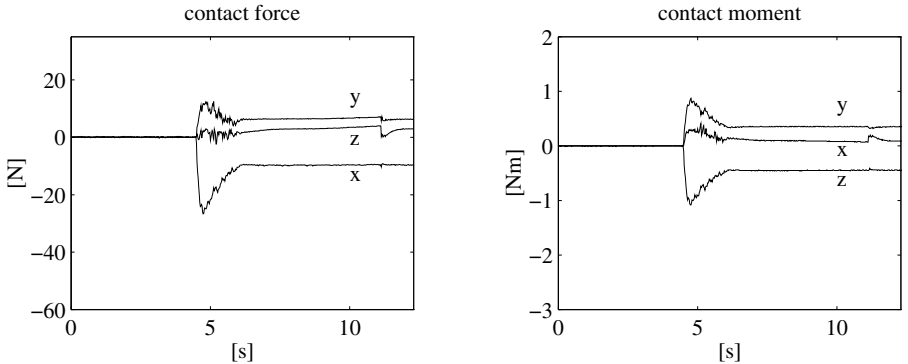


Figure 5.4. Experimental results under six-DOF impedance control based on quaternion in the second case study

The results in Figures 5.4 and 5.5 show the significant differences occurring in the performance of the two schemes. For the impedance control based

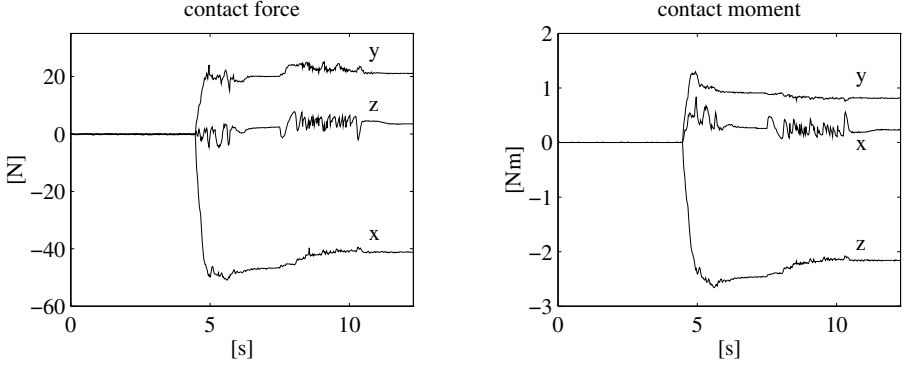


Figure 5.5. Experimental results under six-DOF impedance control based on the classical Euler angles in the second case study

on (3.3), large values of contact force and moment are generated since the rotational impedance equation suffers from ill-conditioning of the matrix $\mathbf{T}(\varphi_c)$; this phenomenon is not present for the quaternion-based impedance control based on (3.30) since representation singularities are not involved in the rotational impedance equation. On the other hand, testing of the impedance control based on the alternative Euler angles in (3.6) has revealed a performance as good as the quaternion-based impedance control, since the orientation displacement φ_{dc} is kept far from a representation singularity. Hence, the results are not reported here for brevity.

In sum it can be concluded that both the impedance control based on the alternative Euler angles and the quaternion-based impedance control perform better than the impedance control based on the classical Euler angles, as far as interaction with the environment is concerned.

Third case study: Task geometric consistency. Another case study has been developed to analyse task geometric consistency when an external moment is applied at the end effector. The quaternion-based impedance control and the impedance control based on Euler angles have been tested.

The stiffness matrices of the rotational impedance equations (3.30), (3.3) have been taken as diagonal matrices; $\mathbf{K}_o = \mathbf{U}_o^T \boldsymbol{\Gamma}_o \mathbf{U}_o$ has been chosen with $\mathbf{U}_o = \mathbf{I}$ and $\boldsymbol{\Gamma}_o = 2.5\mathbf{I}$ for both schemes. The remaining parameters of the rotational impedance have been set to $\mathbf{M}_o = 0.25\mathbf{I}$ and $\mathbf{D}_o = 1.5\mathbf{I}$ for both schemes. No impedance control has been accomplished for the translational part. The gains of the inner motion control loop have been chosen equal to those in the previous case study.

The position and orientation of the desired frame are required to remain constant, and a torque is applied about the approach axis of Σ_d ; the torque is taken from zero to 2.5 Nm according to a linear interpolating polynomial with 4th-order blends and a total duration of 1 s.

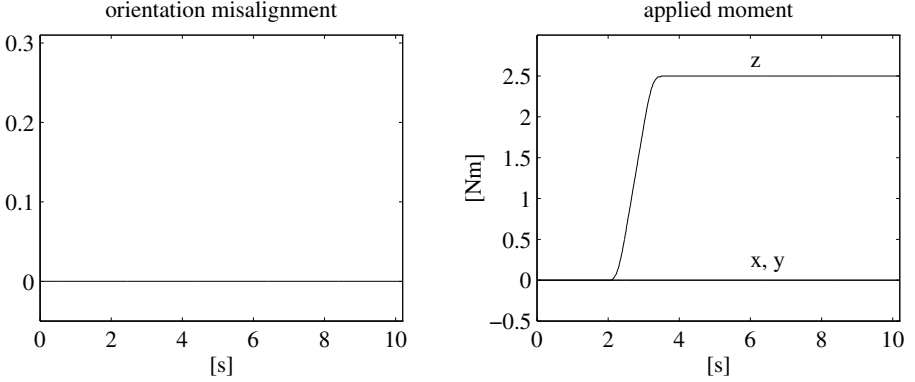


Figure 5.6. Experimental results under six-DOF impedance control based on quaternion in the third case study

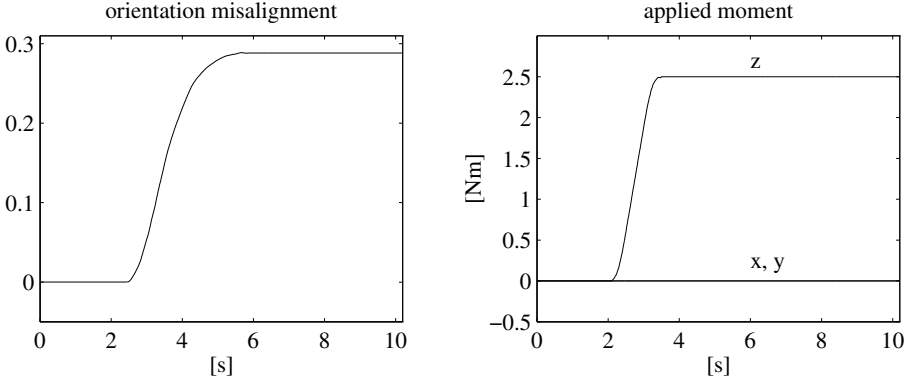


Figure 5.7. Experimental results under six-DOF impedance control based on classical Euler angles in the third case study

The results in Figures 5.6 and 5.7 show the different performance in terms of the orientation misalignment δ which has been defined as the norm of the vector product between the orientation error and the unit vector \mathbf{u}_{o3} of the approach axis of Σ_d , i.e.

$$\delta = \|\mathbf{S}({}^e\epsilon_{de})\mathbf{u}_{o3}\|.$$

For the impedance control based on (3.3) the instantaneous axis of rotation of Σ_e changes, while remarkably no misalignment occurs for the impedance control based on (3.30). The impedance control based on (3.6) has also been tested and its performance is as good as that of the quaternion-based control; hence, the results are not reported for brevity.

Fourth case study: Nondiagonal rotational stiffness. In the fourth case study, the quaternion-based impedance control and the impedance control based on

the alternative Euler angles have been tested when a nondiagonal rotational stiffness is chosen. The impedance control based on the classical Euler angles has been ruled out in view of the poor results of the previous experiment.

The principal axes of the stiffness matrices of the rotational impedance equations (3.30),(3.6) are rotated with respect to the coordinate axes of Σ_d ; $\mathbf{K}_o = \mathbf{U}_o^T \mathbf{\Gamma}_o \mathbf{U}_o$ has been chosen with

$$\mathbf{U}_o = \begin{bmatrix} 0.8047 & -0.3106 & 0.5059 \\ 0.5059 & 0.8047 & -0.3106 \\ -0.3106 & 0.5059 & 0.8047 \end{bmatrix} \quad \mathbf{\Gamma}_o = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 2.5 \end{bmatrix}$$

for both schemes. The remaining parameters of the rotational impedance have been set to $\mathbf{M}_o = 0.25\mathbf{I}$ and $\mathbf{D}_o = 1.5\mathbf{I}$ for both schemes. As above, no impedance control has been accomplished for the translational part, and the gains of the inner motion control loop have been chosen equal to those in the previous case study. A torque has been applied about the axis whose unit vector is \mathbf{u}_{o3} ; the torque is taken from zero to -1.5 Nm according to a linear interpolating polynomial with 4th-order blends and a total duration of 1 s.

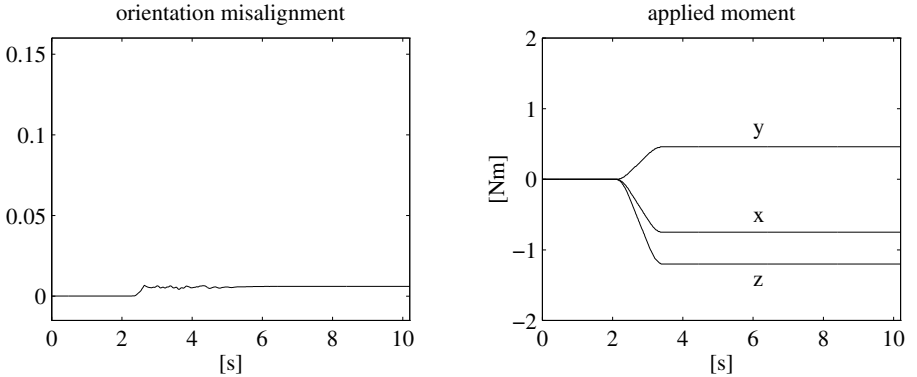


Figure 5.8. Experimental results under six-DOF impedance control based on quaternion in the fourth case study

The results in Figures 5.8 and 5.9 show the significant differences occurring in terms of the orientation misalignment δ . It can be seen that the instantaneous axis of rotation of Σ_e does not appreciably rotate with the impedance control based on (3.30), given the performance of the inner loop acting on the end-effector orientation error. Instead, a significant misalignment occurs with the impedance control based on (3.6).

In sum, it can be concluded that the quaternion-based impedance control performs better than both impedance control schemes based on the Euler angles as far as task geometric consistency is concerned.

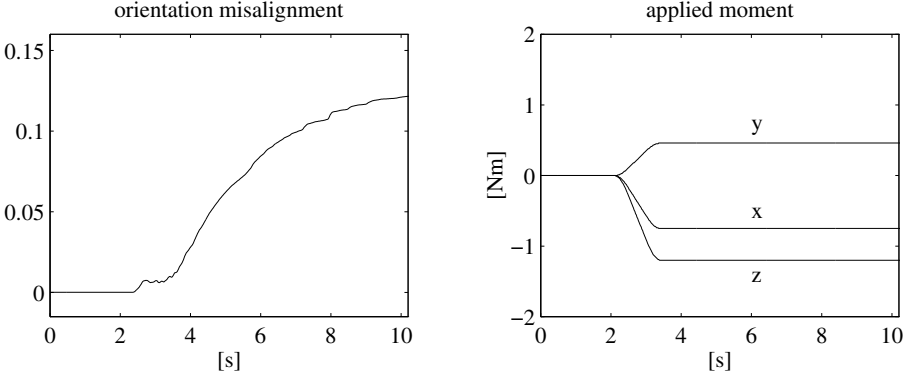


Figure 5.9. Experimental results under six-DOF impedance control based on alternative Euler angles in the fourth case study

Fifth case study: Redundancy resolution. The six-DOF quaternion-based impedance control has been tested in a case study when a redundancy resolution scheme is incorporated into the motion control.

The environment is constituted by a cardboard box. The translational stiffness at the contact between the end effector and the surface is of the order of 5000 N/m, while the rotational stiffness for small angles is of the order of 15 Nm/rad.

The task in the experiment consists of four phases; namely, reconfiguring the manipulator, approaching the surface, staying in contact, and leaving the surface. To begin, the additional task function in (3.39) has been chosen as

$$w(\mathbf{q}) = \frac{1}{2}(q_3 - q_{3d})^2$$

where q_3 is the elbow joint and q_{3d} is a desired trajectory from the initial value of q_3 to the final value of 1.1 rad in a time of 4 s with a fifth-order interpolating polynomial with null initial and final velocity and acceleration. This function is aimed at reconfiguring the manipulator in a more dexterous posture before contacting the surface. After a lapse of 4 s, the disk is taken in contact with the surface at an angle of $7\pi/36$ rad. The end-effector desired position is required to make a straight-line motion with a horizontal displacement of 0.08 m along the Z_b axis of the base frame. The trajectory along the path is generated according to a fifth-order interpolating polynomial with null initial and final velocities and accelerations, and a duration of 2 s. The end-effector desired orientation is required to remain constant during the task. The surface is placed (vertically) in the X_bY_b -plane of the base frame in such a way to obstruct the desired end-effector motion, both for the translational part and the rotational part. After a lapse of 13 s in contact, the end-effector motion is commanded back to the initial position with a duration of 4 s.

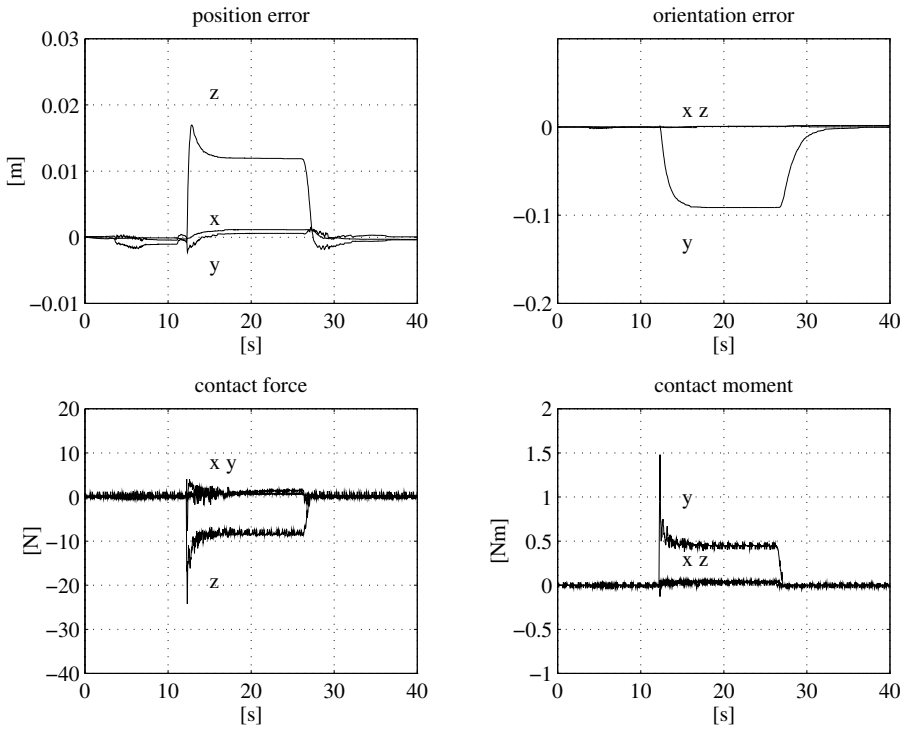


Figure 5.10. Experimental results under six-DOF impedance control based on quaternion with redundancy resolution in the fifth case study

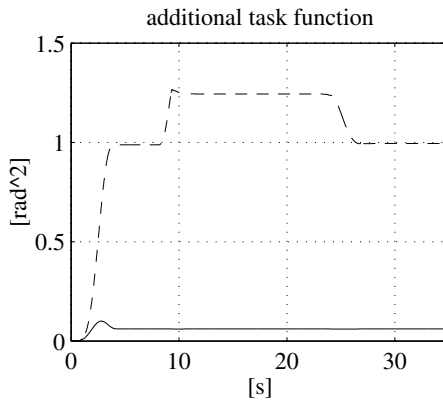


Figure 5.11. Additional task function in the fifth case study

The parameters of the translational impedance (3.1) have been set to $\mathbf{M}_p = 16\mathbf{I}$, $\mathbf{D}_p = \text{diag}\{800, 800, 250\}$ and $\mathbf{K}_p = \text{diag}\{1300, 1300, 800\}$, while the parameters of the rotational impedance (3.30) have been set to $\mathbf{M}_o = 0.7\mathbf{I}$ and $\mathbf{D}_o = 4\mathbf{I}$, $\mathbf{K}_o = 2.5\mathbf{I}$.

The gains of the inner motion control loop in (2.9),(2.15) have been set to $\mathbf{K}_{Pp} = 2250\mathbf{I}$ and $\mathbf{K}_{Po} = 4000\mathbf{I}$, $\mathbf{K}_{Dp} = 70\mathbf{I}$ and $\mathbf{K}_{Do} = 75\mathbf{I}$. The gains of the redundancy resolution control in (3.38),(3.39) have been set to $\mathbf{K}_n = 20\mathbf{I}$ and $k_\beta = 250$.

The results in Figure 5.10 show the effectiveness of the six-DOF impedance control with redundancy resolution. During the reconfiguration (8 s), the components of the position error $\Delta\mathbf{p}_{de} = \mathbf{p}_d - \mathbf{p}_e$ between Σ_d and Σ_e and of the orientation error ${}^e\epsilon_{de}$ between Σ_d and Σ_e are practically zero, meaning that the dynamics of the null space motion does not disturb the end-effector motion. Such error remains small during the approach (2 s). During the contact (13 s), the component of the position error along the Z_b -axis significantly deviates from zero, as expected; as for the orientation error, the component of the orientation error along the Y_e -axis significantly deviates from zero since Σ_e has to rotate about Y_e in order to comply with the surface. Also, in view of the imposed task, a prevailing component of the contact force can be observed along the Z_b -axis after the contact, whereas the sole component of the contact moment about the Y_e -axis is significant, as expected. During the takeoff (4 s), both the errors and the contact force and moment return to zero.

The same task has been executed again for the impedance control without redundancy resolution ($k_\beta = 0$). The performance in terms of the contact between the end effector and the surface is the same as above since the additional task does not interfere with the primary interaction task; hence, the time history of the relevant quantities is omitted for brevity. Nevertheless, a comparison between the two cases in Figure 5.11 shows that the task function is successfully optimised when redundancy is exploited (solid) other than when redundancy is not exploited (dashed).

5.2 Loose Cooperation

A typical assembly task has been performed to test the effectiveness of the quaternion-based impedance control strategy for execution of interaction tasks where the geometry of the contact plays a crucial role. The task is the classical peg-in-hole, which can be executed by a dual-arm system if one robot carries the peg and the other holds the hollow part.

The task for the seven-joint manipulator is planned as follows. From a given posture a joint space motion is commanded to reach a suitable intermediate posture which facilitates the subsequent phases of the task; then, a Cartesian space motion along a straight-line path is commanded to drive the tip of the peg in the proximity of the mouth of the hole and align the

approach axis of the peg with the axis of the hole, as accurately as possible. Finally, a straight-line motion along the approach axis —typically at a reduced speed with respect to the previous phase— is commanded to realise the insertion. The described task is specified in the following PDL 2 program:

```

PROGRAM insertion CONST spd=3
      dpt=50
VAR   int: JOINTPOS
      pro: POSITION
BEGIN
      --move joints to intermediate posture
      MOVE TO int
      --move tip to proximate pose
      MOVE LINEAR TO pro
      --set arm speed override at spd %
      $ARM_OVR := spd
      --move tip along approach axis by dpt mm
      MOVE RELATIVE VEC(0,0,dpt) IN TOOL
END insertion

```

where the joint position **int** and the end-effector pose **pro** are taught beforehand, while the constants **spd** and **dpt** are set according to task requirements.

The six-joint manipulator is controlled using the open operating mode so that the end effector behaves as the programmed six-DOF impedance. To this purpose, the force/torque sensor is mounted at the wrist.

For the task at issue, the position and orientation of the desired frame is taken as a constant, i.e. the six-joint manipulator is controlled to stay still. Whenever a contact force and/or moment is experienced at the end effector, this reacts according to the programmed impedance where the origin of the desired frame determines the location of the Remote Center of Compliance.

The peg is a wooden cylinder of 17 mm diameter and 80 mm height, while the hollow part is a wooden block with a hole of 18 mm diameter and 70 mm depth; that is, a 0.5 mm radial tolerance is present during the insertion.

The insertion task is programmed in terms of a planned motion for the seven-joint manipulator described by **spd** = 3 and **dpt** = 50.

The six-joint manipulator is impedance-controlled so that the Remote Center of Compliance is located at $\mathbf{p}_d = \mathbf{p}_0 - [0 \ 165 \ 0]^T$ mm since the insertion direction is along the Y_b -axis of the base frame. Also, it is $\mathbf{R}_d = \mathbf{R}_0$. The parameters of the impedance equation in (3.1),(3.30) are respectively set to $\mathbf{M}_p = \text{diag}\{15, 40, 15\}$, $\mathbf{D}_p = \text{diag}\{300, 950, 300\}$, $\mathbf{K}_p = \text{diag}\{400, 1300, 400\}$, for the translational part, and $\mathbf{M}_o 9\mathbf{I}$, $\mathbf{D}_o = 13.5\mathbf{I}$, $\mathbf{K}_o = \mathbf{I}$, for the rotational part.

Since the experiment is aimed at testing the robustness of the proposed strategy in the case of incorrect task planning, an additional misalignment has been intentionally introduced by rotating the end-effector frame of the seven-joint manipulator by 2 deg about the X_b - and the Z_b -axis of its base

frame. Further, the length of the path along the approach axis of the peg has been set to $d_{pt} = 90$, corresponding to a 20 mm overshoot beyond the bottom of the hole. The same impedance parameters as above are chosen.

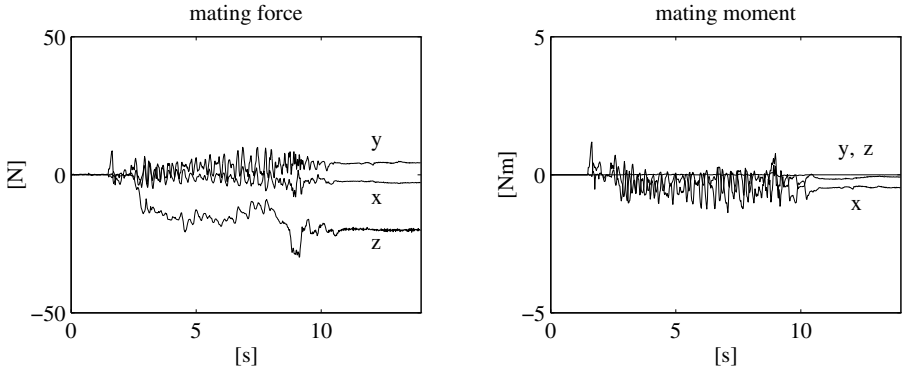


Figure 5.12. Force and moment in the peg-in-hole task

The results are illustrated in Figure 5.12 in terms of the time history of the three components of mating force and moment. Interestingly enough, the values of force and moment keep limited despite of the incorrect task planning; moreover at steady state, nonnull values of force and moment can be observed which are obviously caused by the planned misalignment and overshoot.

5.3 Tight Cooperation

In the experiment devoted to testing the proposed tight cooperative control strategy, the two robot end effectors tightly grasp the ends of a wooden bar of 1 m length. At the center of the bar is fixed a steel stick with a wooden disk of 5.5 cm radius at its tip.

The environment is constituted by a cardboard box; the translational stiffness at the contact between the disk and the surface is of the order of 5000 N/m, while the rotational stiffness for small angles is of the order of 15 Nm/rad.

The task in the experiment consists in taking the disk in contact with the surface; that is placed at an unknown distance with an angle of unknown magnitude. The origin of Σ_o is required to make a desired motion along a straight line with a vertical displacement of -0.275 m along the Z_b -axis of Σ_b . The trajectory along the path is generated according to a 5th-order interpolating polynomial with null initial and final velocities and accelerations, and a duration of 6 s. The desired orientation of the object frame is required to remain constant.

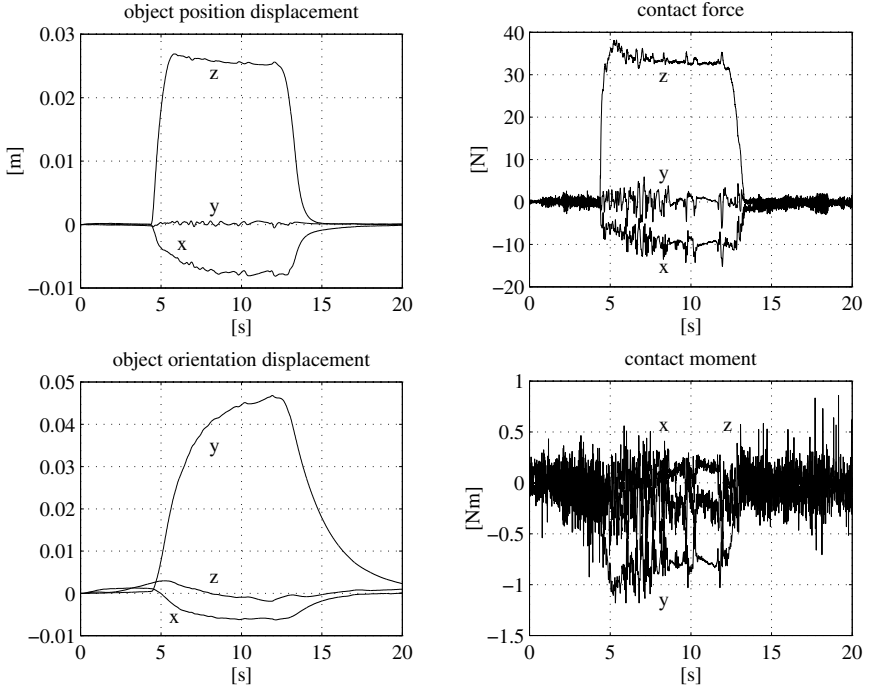


Figure 5.13. Experiment of tight cooperative control: *Left:* Object position and orientation displacement between Σ_d and Σ_o . *Right:* Contact force and moment acting on the object

The parameters of the translational part of the impedance equation (3.1) have been set to $\mathbf{M}_p = 30\mathbf{I}$, $\mathbf{D}_p = 555\mathbf{I}$, $\mathbf{K}_p = 1300\mathbf{I}$, while the parameters of the rotational part of the impedance equation (3.30) have been set to $\mathbf{M}_o = \text{diag}\{10, 2, 10\}$, $\mathbf{D}_o = \text{diag}\{35, 20, 35\}$, $\mathbf{K}_o = \text{diag}\{20, 8, 20\}$. Notice that the stiffness matrices have been chosen so as to ensure a compliant behaviour (limited values of contact force and moment) during the contact, while the damping matrices have been chosen so as to guarantee a well-damped behaviour. The feedback gains in (2.9, (2.15) have been set to $k_{Dp} = 65$, $k_{Pp} = 1800$ for the position loop, and $k_{Do} = 65$, $k_{Po} = 3600$ for the orientation loop, respectively.

From Figure 5.13 (*left*), after the contact, the component along the Z_b -axis of the position displacement between the desired frame Σ_d and the object frame Σ_o , expressed in Σ_b , significantly deviates from zero, as expected; a smaller displacement can also be seen for the component along the X_b -axis, due to contact friction. As for the orientation displacement between Σ_o and Σ_d , expressed in Σ_d , only the component along the Y_d -axis significantly deviates from zero since the object frame has to rotate about the Y_d -axis of Σ_d in order to comply with the surface after the contact.

From Figure 5.13 (*right*), in view of the imposed task, a prevailing component of the contact force can be observed along the Z_b -axis after the contact, while a significant component along the X_b -axis arises, corresponding to the above position displacement. As for the contact moment, the only nonnegligible component is that along the Y_d -axis of Σ_d , which corresponds to the above orientation displacement. It can be recognised that all the above quantities reach constant steady-state values after the desired motion is stopped. The oscillations on the force and moment can be ascribed to the flexibility effects of the commonly held object.

6 Conclusion and Future Directions

An impedance control strategy has been presented for six-DOF tasks. By considering an angle/axis representation of the end-effector orientation displacement, a six-DOF impedance equation has been derived, which exhibits a physically meaningful behavior and task geometric consistency properties. Among the different angle/axis representations, the unit quaternion has been chosen which avoids the occurrence of representation singularities. The superior performance of the proposed angle/axis-based impedance controller over two different impedance controllers based on Euler angles has been shown both in theory and in practice.

For the case of redundant manipulators, the dynamically consistent pseudoinverse of the manipulator Jacobian has been adopted to decouple the dynamics of the end-effector motion from the null-space motion. Moreover, redundancy is exploited to stabilise null-space joint velocities and optimise an additional task function.

Finally, the problem of two cooperative robots manipulating a common object has been considered and both loose and tight cooperation have been addressed. In the case of loose cooperation, limited interaction forces have been achieved by adopting the geometrically consistent impedance control strategy for one of the two manipulators. In the case of tight cooperation, interaction of the grasped object with the environment has been managed by enforcing a geometrically consistent impedance behaviour at the object level.

The proposed control strategies have been experimentally tested on a set-up composed by two industrial robots Comau SMART-3S with two force/torque sensors ATI FT-30/100 mounted at the wrist.

Future work will be devoted to extending the use of geometrically consistent impedance for cooperative manipulators. Namely, the adoption of an impedance control strategy for controlling the internal forces at the grasped object is currently under investigation. Under this regard, an open problem is the cooperative manipulation of flexible objects. It is expected that the adoption of impedance control may provide a simple and robust way to cope with manipulation of flexible objects or manipulation of rigid objects via elastic grasp.

Promising future research will be directed towards the integration of the interaction control strategies, based on the use of position and force/torque sensors, with information coming from vision sensors. Indeed, the adoption of sensory feedback coming from cameras is expected to enhance the robustness of the control loop as well as may represent a decisive step towards robotic systems capable of autonomous interaction with the surrounding environment, even partially.

Last but not least, it should be mentioned that the problem of ensuring a safe interaction between the robot the environment can be considered as a major objective to be pursued in the next few years. In the case of interaction with an artificial environment, suitable strategies must be adopted aimed at preventing and/or managing robot failures and abrupt changes of the environment conditions, so as to avoid damage for the various devices involved in the contact and ensure task completion. On the other hand, when the human/robot interaction is of concern (e.g. in anthropic and service robotics), safety of the human subject becomes the primary objective to be achieved, even at the expense of the task assigned to the robot.

Acknowledgements

The research activity of the PRISMA group has been cofunded by *Italian Space Agency* (ASI) under grant ARS98-190.

References

1. Bruni F, Caccavale F, Natale C, Villani L 1996 Experiments of impedance control on an industrial robot manipulator with joint friction. In: *Proceedings of 1996 IEEE International Conference on Control Applications* Dearborn, MI, pp 205–210
2. Caccavale F, Chiacchio P 2000 An experimental setup for cooperative manipulation based on industrial manipulators. *Industrial Robot* 27:120–130
3. Caccavale F, Chiacchio P, Chiaverini S 2000 Task-space regulation of cooperative manipulators. *Automatica* 36:879–887
4. Caccavale F, Chiaverini S, Natale C, Siciliano B, Villani L 2000 Geometrically consistent impedance control for dual-robot manipulation. In: *Proceedings of 2000 IEEE International Conference on Robotics and Automation* San Francisco, CA, pp 3873–3878
5. Caccavale F, Natale C, Siciliano B, Villani L 1998 Control of two industrial robots for parts mating. In: *Proceedings of 1998 IEEE International Conference on Control Application* Trieste, I, pp 562–566
6. Caccavale F, Natale C, Siciliano B, Villani L 1998 Resolved-acceleration control of robot manipulators: A critical review with experiments. *Robotica* 16:565–573
7. Caccavale F, Natale C, Siciliano B, Villani L 1999 Six-DOF impedance control based on angle/axis representations. *IEEE Transactions on Robotics and Automation* 15:289–300

8. Caccavale F, Siciliano B, Villani L 1998 Robot impedance control with nondiagonal stiffness. *IEEE Transactions on Automatic Control* 44:1943–1946
9. Chiacchio P, Chiaverini S, Siciliano B 1996 Direct and inverse kinematics for coordinated motion tasks of a two-manipulator system. *ASME Journal of Dynamic Systems, Measurement, and Control* 118:691–697
10. Chiaverini S, Sciacivco L, 1993 The parallel approach to force/position control of robotic manipulators. *IEEE Transactions on Robotics and Automation* 9:361–373
11. Cutkosky M R, Wright P K 1986 Active Control of a Compliant Wrist in Manufacturing Tasks. *ASME Journal of Dynamic Systems, Measurement, and Control* 108:36–43
12. De Schutter J, Van Brussel H 1988 Compliant robot motion II. A control approach based on external control loops. *International Journal of Robotics Research* 7(4):18–33
13. Featherstone R, Khatib O 1997 Load independence of the dynamically consistent inverse of the Jacobian matrix. *International Journal of Robotics Research* 16:168–170
14. Hogan N 1985 Impedance control: An approach to manipulation: Parts I—III. *ASME Journal of Dynamic Systems, Measurement, and Control* 107:1–24
15. Hollis R L, Salcudean S E, Allan A P 1991 A six-degree-of-freedom magnetically levitated variable compliance fine-motion wrist. *IEEE Transactions on Robotics and Automation* 7:320–332
16. Hsu P, Hauser J, Sastry S 1989 Dynamic control of redundant manipulators. *Journal of Robotic Systems* 6:133–148
17. McClamroch N H, Wang D 1988 Feedback stabilization and tracking of constrained robots. *IEEE Transactions on Automatic Control* 33:419–426
18. Mills J K, Goldenberg A A 1989 Force and position control of manipulators during constrained motion tasks. *IEEE Transactions on Robotics and Automation* 5:30–46
19. Nakamura Y 1991 *Advanced Robotics: Redundancy and Optimization* Addison-Wesley, Reading, MA
20. Natale C, Siciliano B, Villani L 1999 Spatial impedance control of redundant manipulators. In: *Proceedings of 1999 IEEE International Conference on Robotics and Automation* Detroit, MI, pp 1788–1793
21. Paul R, Shimano B 1976 Compliance and control. In: *Proceedings of 1976 Joint Automatic Control Conference* San Francisco, CA, pp 694–699
22. Raibert M H, Craig J J 1981 Hybrid position/force control of manipulators. *ASME Journal of Dynamic Systems, Measurement, and Control* 103:126–133
23. Salisbury J K 1980 Active stiffness control of a manipulator in Cartesian coordinates. In: *Proceedings of 19th IEEE Conference on Decision and Control* Albuquerque, NM, pp 95–100
24. Sciacivco L, Siciliano B 2000 *Modelling and Control of Robot Manipulators* (2nd ed) Springer, London, UK
25. Siciliano B, Villani L 1999 *Robot Force Control* Kluwer, Boston, MA
26. Uchiyama M, Dauchez P 1993 Symmetric kinematic formulation and non-master/slave coordinated control of two-arm robots. *Advanced Robotics* 7:361–383
27. Yoshikawa T 1987 Dynamic hybrid position/force control of robot manipulators—Description of hand constraints and calculation of joint driving force *IEEE Journal of Robotics and Automation* 3:386–392
28. Whitney D E 1982 Quasi-static assembly of compliantly supported rigid parts. *ASME Journal of Dynamic Systems, Measurement, and Control* 104:65–77

Impact Modelling and Control of Robotic Links

Marina Indri

Dipartimento di Automatica e Informatica
Politecnico di Torino
<http://www.polito.it/research/robotics>

Impact modelling and control can be fundamental in robotics, for the execution of tasks that require the interaction of the robot with an external body, ensuring that a stable contact is achieved without rebounds. This chapter examines various issues related to impact modelling and control, devoting a particular attention to the main results of the last decade. After the illustration of some general approaches to impact modelling in robotics applications, various impact control schemes are surveyed, highlighting the possible problems that can limit in practice their performances. For this reason, particular attention is devoted to those control schemes whose performances have been also experimentally tested. A general classification between *switching* and *non-switching* impact control schemes is introduced, and some control solutions, developed by the author following a non-switching approach, are illustrated and discussed.

1 Introduction

The most recent robotics applications are not restricted to the classical fields, such as the industrial and the manufacturing ones, but they are relative to various contexts, e.g. as biomedical services, operations in hostile environments, and entertainment, too.

The most common control schemes, which are widely used in the industrial field, are only partially suitable for such new applications, as they are mainly constituted by *position* control schemes, while several among these new tasks require the interaction of the robot with some external object. During the last years, many research studies about robot interaction control have been developed: some of the most recent and important results are presented and discussed in a devoted chapter of this monograph. But several works about force and impedance control (see the numerous references in the previous chapter) deal with the control of robot interaction with an external environment, only once a stable contact situation has been already, somehow, achieved. In practice, this can be difficult to be obtained with no rebounds without an explicit control of the transition phase of the robot, from the free motion condition to the constrained one. *Impact* control can then be fundamental not only to preserve the mechanical structure of the robot from

failures, which can be generated when impact forces are not properly recognised and taken under control, but also as a control goal, for the execution of tasks in which the robot transition toward a contact situation (with an external object or with another cooperating robot) must be performed without rebounding and loss of contact.

The first matter to be analysed is the construction of a suitable impact model for control. The basic physical phenomena that occur when two or more bodies collide have been widely studied and analysed in several books, such as e.g. [1], [7], [17], [30]. In the development of the dynamic model of a robot striking an external object, it is fundamental to properly choose a satisfying compromise between a too detailed description of impact from a mechanical and structural point of view (to avoid too complex equations), and an over-simplified model, which cannot adequately explain those instability and bouncing problems that may occur in practice, and that must be afforded in the design of the control scheme. Interesting works appeared in the last decade about impact modelling in robotics (see e.g. [1], [9], [10], [22], [25], [29]), as it will be discussed in Section 2.

No general theory is available for impact control, because the dynamic equations, which can adequately describe the impact of a robot with an external environment, strongly depend on the geometry of both the manipulator and the environment, and on their physical and structural properties, which may be unknown or only partially known. Due to the sudden change of the equations of motion, that occurs when the bodies involved in the impact switch sharply from a condition of non-contact to a condition of contact, the presence of some unexpected disturbances can make the manipulator leave the contact surface after impact, and limit cycles or instabilities may be excited. Besides, quite powerful tools are required for the practical application of many impact control schemes, e.g. for precise data storing and/or for the impact detection: experimental tests are then fundamental to validate the actual effectiveness and performances of the developed impact control strategies.

A survey of the most recent results obtained for impact control in the robotic field is presented in Section 3, by introducing a general classification of them with reference to the existence or not of a “switching condition” in the definition of the control scheme, i.e. by distinguishing if a unique control law is applied during free-motion, contact mode and in the impact transition phase between them, or if different controllers are employed in the three situations, by switching from a control law to another one. A particular attention is devoted to the employment of integral terms in the switching control laws, as such terms, which are crucial for force control in the contact condition, can determine bouncing and instability during the transition phase.

In Section 4 some impact control solutions, which have been developed by the author together with Tornambè (see [12]–[16]), with the aim of studying the possibility to achieve satisfying results by the application of a unique

control law (both in free motion and during contact), are illustrated and discussed on the basis of some experimental results, too.

Section 5 draws some final conclusions.

2 Impact Modelling

A *good* impact model of a robot, striking some external environment, must be the result of a satisfying compromise: in fact, it must be quite simple, to be suitable for the subsequent controller design, but at the same time it must correctly take into account the various phenomena involved in the impact itself. Some more or less restrictive assumptions must then be made to *a priori* establish the approximation level to be used to describe impact. In particular, assumptions must be made about the possibility of neglecting or not plastic deformations of the structures of the involved systems, about the presence or not of friction between the impacting surfaces, and about the importance of rotational effects, in order to simplify the model, which must correctly describe how the different parameters (such as e.g. masses, geometric dimensions, stiffness, impact duration) affect the collision.

The basic characteristic of the models of impacting systems is given by the existence of two separate conditions, the free motion one and the contact one, together with a transition phase between them (i.e. the actual impact phase). Kinematic and/or dynamic constraints may hold in the contact condition and during the transition phase, which is characterised by a *jump* in the velocities of the systems involved in the impact: the relation between post- and pre-impact velocities is specified by the so-called *restitution rule* [17], [30].

Some of the main results that appeared in literature in the last decade about impact modelling in the robotic field are briefly recalled and discussed hereafter.

2.1 Some “Pioneering” Works

The restitution rule, relating post- and pre-impact velocities of colliding systems, is the basis of the impact model that was developed and experimentally validated in the paper [29], which is to be mentioned among the ones that appeared in the first 90’s about impact modelling in the robotic field. Here, a simple energy approach is used to determine a quite accurate impact model in the case of a single-link manipulator (actuated by a high torque DC servo motor, geared down by a speed reducer), whose tool comes into contact with a workpiece. The model, which takes into account the stiffness of the tool and the workpiece, but also the torsional stiffness of the transmission, is based on the assumption that when the tool strikes the workpiece for the first time, a common impulsive impact force is applied to them for a short time, during which they are still free from each other. Afterwards, when contact is

established, the tool and the workpiece can be modelled as a unique lumped mass. The impact description is then obtained by defining the impulse as the time integral of the force, and by imposing that it equals the change in momentum, expressed as a function of the coefficient of restitution e , defined as:

$$e = \frac{v_{2,f} - v_{1,f}}{v_{1,i} - v_{2,i}}, \quad (2.1)$$

where $v_{k,i}$ and $v_{k,f}$, $k = 1, 2$, are respectively the initial and final velocity components of the colliding object k , in the direction normal to the contact surfaces.

The papers [9] and [10], which appeared too in the first 90's, deal with rigid bodies collisions of planar kinematic chains, as in the case of walking robots and locomotion systems, which are subject to periodic impacts with the walking surface.

In particular, the work in [9] deals with the class of kinematic chains having one end resting on the ground surface prior to impact, while the other end contacts the surface (just as in the locomotion case); a methodology is proposed, and applied to some examples, to exactly determine the post impact velocities of the kinematic chains, when they strike external surfaces in presence of friction.

Paper [10] is devoted to the solution of *multicontact* collisions of kinematic chains. Two procedures are developed to solve the problem: the first one, which is based on a *differential* formulation of the impact equations, allows the computation of the post-impact velocities accordingly to three different possible definitions of the coefficient of restitution (the kinematic, the kinetic, and the energy one); the second procedure uses the kinematic definition of the coefficient of restitution only, on the basis of an *algebraic* formulation of the impact equations (see [10] for details). This last formulation cannot be applied to the cases in which rebounds occur at the interaction with the surface. The application of the different procedures to a specific example, given by a planar three-link chain with two contact points, is used to discuss the consistency of the obtained results using the various approaches, and the possibility to predict rebounds at the non-impacting contact points.

A common weak point of these “old” works is represented by the fact that they deal with particular cases or classes of systems, so that their results can be only partially extended directly to more general applications.

2.2 General Approaches to Impact Modelling

A formal, well stated discussion about modelling (and control) of impacting systems can be found in [2] (and in a more complete way in the book [1]), where a general formulation is proposed for the model of rigid manipulators subject to impacts. In particular, the three phases (free motion, contact, and transition between them) are defined as recalled hereafter.

Let $q \in \mathbb{R}^n$ be the vector of the generalised coordinates of an n -DOF rigid robot, and let $F(q) = 0$ be the constraint equation, describing a surface with which the manipulator must interact. In the most general case, $F(q) \in \mathbb{R}^m$ with $m \geq 1$, i.e. multi-dimensional constraints can be considered; in the following developments, a codimension-one constraint will be assumed for the sake of simplicity. The well-known robot dynamic equations in the free-motion phase and in the constrained-motion one can then be written as follows:

Free-motion phase

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = U \quad (2.2)$$

$$F(q) > 0. \quad (2.3)$$

Constrained-motion phase

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = U + \lambda \frac{\partial F}{\partial q}(q) \quad (2.4)$$

$$F(q) = 0 \quad \text{and} \quad \lambda \geq 0. \quad (2.5)$$

where $M(q) \in \mathbb{R}^{n \times n}$ is the positive definite inertia matrix, $C(q, \dot{q})\dot{q}$ contains the Coriolis and centrifugal terms, $g(q) \in \mathbb{R}^n$ is the vector of the generalised gravity torques, $U \in \mathbb{R}^n$ is the control input vector (to be defined accordingly to the required task), and λ is the Lagrange multiplier.

In the first case the robot motion is free from obstacles, and it is described by the set of ordinary differential equations given in (2.2). In the classical interaction and hybrid force/position control schemes, the robot is generally supposed to be already in the constrained-motion condition described by (2.4)-(2.5); the constraint $F(q) = 0$ is assumed to hold for all times, without any consideration for possible transitions between robot configurations such that $F(q) > 0$ and other ones such that $F(q) = 0$. In this way, the system trajectories can still be considered as smooth time functions.

To properly consider impact, the robot dynamic model given by (2.2)–(2.5) must be completed with the description of the transition phase between the free-motion condition and the constrained-motion one. Let $t = t_c$ be the time instant at which a collision occurs; it follows that:

- $F[q(t)] > 0$ for $t \in [t_c - \delta, t_c]$, for some $\delta > 0$ small enough, and $F[q(t_c)] = 0$;
- the velocity $\dot{q}(t)$ has a discontinuity at $t = t_c$, such that

$$\dot{q}(t_c^-)^T \nabla_q F[q(t_c)] < 0 \quad \text{and} \quad \dot{q}(t_c^+)^T \nabla_q F[q(t_c)] \geq 0,$$

where $\dot{q}(t_c^-)$ and $\dot{q}(t_c^+)$ are, respectively, the left- and the right-limit of the velocity $\dot{q}(t)$ for $t \rightarrow t_c$, and $\nabla_q F[q(t_c)]$ is the gradient of $F(q)$ at $t = t_c$; the jump in \dot{q} , denoted as $\sigma_{\dot{q}}(t_c)$, can be specified through the restitution rule relating post- and pre-impact velocities;

- due to the discontinuity in the velocity, a *percussion* $P_q(t_c)$ acts on the system at $t = t_c$, i.e. a generalised impulsive force of the form $P_q(t_c)\delta_{t_c}$, where δ_{t_c} is the *Dirac measure* at t_c .

The robot dynamic model (2.2)–(2.5) is then completed by the following equations.

Transition phase

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = U \quad (2.6)$$

$$F(q) > 0 \quad (2.7)$$

$$M[q(t_c)]\sigma_{\dot{q}}(t_c) = P_q(t_c) \quad (2.8)$$

$$F[q(t_c)] = 0 \quad (2.9)$$

$$\dot{q}(t_c^+) = -e\dot{q}(t_c^-), \quad (2.10)$$

where $e \in [0, 1]$ is the *restitution coefficient*, characterising the well-known *Newton's rule* (2.10), which can be extended to higher-dimensional cases by a *generalised normal rule* of the form:

$$\dot{q}(t_c^+)^T \nabla_q F[q(t_c)] = -e\dot{q}(t_c^-)^T \nabla_q F[q(t_c)].$$

It must be stressed that the crucial problem in the impact models, which must be suitable for simulation and control design, is the handling of the inequality constraints on the generalised coordinates $q(t)$, due to the presence of physical obstacles in the robot motion.

A recent study [25] has shown how the equations of motion, resulting from the application of the method of the *Valentine variables*, constitute a suitable way to describe the impact between very stiff mechanical systems (i.e. in case of “nonsmooth” impacts), whereas the method of the *penalty functions* is suitable to describe “smooth” impacts, i.e. impacts of mechanical parts that show some flexibility.

In particular, it is shown how, in case of nonsmooth impacts, the inequality constraints $f_i(q(t)) \leq 0$, $i = 1, 2, \dots, m$, defining the admissible region of $q(t)$ can be transformed into equality constraints of the form:

$$f_i(q(t)) + \gamma_i^2(t) = 0 \quad i = 1, 2, \dots, m \quad (2.11)$$

where $\gamma_i(t) = 0$, $i = 1, 2, \dots, m$ are some auxiliary nonnegative real-valued variables (the so-called *Valentine variables*). The derivation with respect to time of both sides of (2.11) leads to a set of completely equivalent differential constraints, which can be included in the Lagrangian function of the robot (or of the mechanical system in general) by means of m Lagrange multipliers. By looking for the stationary value of the action integral, according to the Hamilton principle, the $n + 2m$ Euler-Lagrange equations of the system are then obtained, and the impact times are determined by the *Erdmann-Weierstrass corner conditions* [25].

The penalty functions method, introduced in the case of smooth impacts, is based on the addition of such functions (one per each constraint) to the expression of the total potential energy of the system (and then to the Lagrangian function that appears in the action integral). Each penalty function takes the zero value when the corresponding constraint is satisfied and a high positive value when the constraint is violated. In this way, the action functional is strongly penalised in correspondence to a violation of the constraints, so that the solution of the modified problem is forced to tend to the actual path of motion of the mechanical system, as the penalisation tends to infinity. Different choices are possible for the penalty functions; the proposed one in [25] is constituted by a sort of fictitious elastic energy, determined by a virtual linear elastic spring at the contact point corresponding to the i th constraint:

$$U_{f_i}(q) := \begin{cases} 0, & \text{if } f_i(q) \leq 0 \\ \frac{1}{2}k_i f_i^2(q), & \text{if } f_i(q) > 0 \end{cases} \quad (2.12)$$

where k_i is a sufficiently high positive number, such that $U_{f_i}(q)$ has the dimension of an energy, when different from zero. An approximation of the actual path of motion of the constrained system is then computed by looking for the unconstrained stationary value of the modified action integral. Some points must be highlighted: (i) the condition number of the problem of computing the stationary value of the action integral is proportional to coefficients k_i , $i = 1, 2, \dots, m$, so that the rate of convergence of any numeric algorithm used to solve the determined equations of motion becomes very slow, as the coefficients k_i are increased in order to obtain more accurate solutions; (ii) only *smooth* impacts can be approximated by the penalty functions method, as the Erdmann-Weierstrass corner conditions obtained in this case imply that the generalised velocities cannot have any jump. More details about these matters can be found directly in [25], where some control laws are also proposed on the basis of the developed impact models.

3 Impact Control in Robotics

Several strategies have been proposed in the most recent years for impact control in robotics applications, but no general theory is available, as the robot and the environment geometries, together with the type of impact, deeply influence not only the form of the equations describing the impacting systems, but also the choice of the most suitable control strategy and its performances.

Some fundamental matters must be taken into account in the design of an impact control strategy, as:

- the available knowledge of kinematics and dynamics of the robotic link and of the external environment with which it interacts (and in particular the knowledge of their stiffness properties);

- the need of detecting the impact time, if the adopted control strategy is constituted by distinct algorithms in the different phases (free-motion, transition, contact);
- the consequent need of measuring the contact forces by means of *suitable* force or deformation sensors, to detect the impact time and/or to implement a force feedback law;
- the quality of the employed force sensors, and in particular their resolution and their dynamic behaviour, to avoid delay in the feedback loop that can cause instability phenomena;
- the possibility of applying alternative control strategies, which do not require the impact detection and/or the insertion of force sensors, when the above problems limit the achievable performances in unacceptable way.

A first, general classification of the several impact control schemes that have been developed can then be made with reference to the existence or not of a switching condition, i.e. with reference to the employment of different controllers in the non-contact and contact phases, or of a unique control law. In the remainder of the section, different impact control schemes are analysed, starting in Section 3.1 from some “old” switching solutions, which were developed about a decade ago, mainly on the basis of experimental tests. The bouncing problems that may occur, especially when integral control laws are employed, are addressed in Section 3.2, where some recent results are reported and discussed. The possibility of using a unique control law for all the phases (free motion, transition, contact), is briefly discussed in Section 3.3, with reference to a scheme that extends the classical impedance control solution, and in Section 4, which is devoted to some non-switching control solutions, developed by the author and Tornambè [12]–[16].

3.1 Some “Old” Experimental Switching Schemes

In the first 90’s, some interesting works appeared about impact control in robotics, presenting some practical and alternative solutions to the classical impedance control schemes that were developed e.g. in [8], [18]. In such papers, the problem was mainly treated from the experimental point of view, by defining switching control strategies that employed different controllers for each phase (free motion, transition, contact). The controllers were generally optimised in some way for the particular phase, in which each of them was applied.

The main contributions provided by such works are constituted by the experimental application of the proposed solutions, and by the carried out analysis of the practical properties and problems of the different controllers.

Among the solutions that were proposed in that period, the control scheme developed by Mills and Lokhorst in [21] should be recalled. Such a scheme is applicable to n -DOF rigid-link robotic manipulators, for the execution of tasks that require transitions from non-contact motion to contact motion

(and vice versa) with a compliant environment, described by a linear lumped-parameter mechanical impedance model. The proposed control strategy is constituted by:

- a PD control law plus a robot dynamics compensation term, during the free motion phase;
- a force/position law, constituted by a PD control law plus a proportional force error term plus a robot dynamics compensation term, during the contact phase;
- an indeterminate control input, which is somewhere between the switching limits, during the transition phase (in which contact is established with a zero contact force).

The environment compliance is fundamental for the performances of such a scheme, which requires the knowledge of the dynamic characteristics of both the robot and the environment, under the assumption that no friction is present at the contact point.

The same three different stages of the contact task (i.e. free motion, impact, and contact after impact) are distinguished in the analysis developed in [20]. Here, an experimental comparison of the different control schemes that can be used is made for the three stages. The best strategies are combined into a single unified controller, basically constituted by a PID controller (based on the position error during the free motion phase, and on the force error during the contact phase), whose gains are tuned by a knowledge-based tuner that makes use of the reference input and the actual position/force information. Good experimental results are presented, but no general discussion of the stability properties of such a scheme is provided: the main difference with respect to the scheme developed in [21] is represented by the introduction of an integral term, which is crucial once contact is established in a stable way, but which can determine rebounds during the impact phase, as it will be discussed with some details in the next subsection.

3.2 Integral Control Laws and Bouncing Problems

It is well-known that integral control, which is the best choice for force control in an established contact situation (see e.g. [28] and [29]), can determine bouncing and instability during the transition phase, because of the integrator wind-up, as it has been discussed in [27], where a switching impact control strategy, constituted by the following three types of control laws, has been proposed:

- Position control is used for free space motion.
- Proportional gain explicit force control with negative gain (indicated as *impact control*) is used to suppress oscillations and bouncing, during the transient phase of the robot impact with the environment.

- Integral gain explicit force control is used to track forces, once stable contact with the environment has been established.

Rebounds are avoided by using different solutions to handle the transitions between the three controllers: the transition from position control to impact control is abrupt and triggered by the impact force spike, whereas the transition from impact control to integral force control is done by a linear switch of all the gain values, after impact force pulse diminishes (the transition times are to be determined empirically).

More recently, the bouncing problem due to the integrator wind-up has been addressed in [6]; this work shows how integral control can be effectively employed for impact control, if some expedients are used. The proposed control law is of the following type:

$$\tau = K_f \int (f - \bar{f}) dt + K_f x_0 - K_v \dot{q}, \quad (3.1)$$

where τ is the motor torque, f is the measured force, \bar{f} is the force set-point, x_0 is the state of the integrator at the detection of the impact, \dot{q} is the motor velocity, K_f and K_v are the integral and velocity feedback gains. Bouncing can be avoided : (i) by choosing a high value for the force set-point (at least during the force spike due to impact), as in this way there is no growth of the wind-up charge; (ii) by considering an appropriate value for the initial integrator state x_0 (an empirical expression of x_0 is given in the paper); (iii) by reducing the integral gain when the contact force vanishes, to avoid other rebounds after a first possible loss of contact.

The negative effects that the time delays, induced in the feedback control loop, can determine when impact is concerned, have been recently studied from the theoretical point of view in [22], in the cases of proportional and proportional-integral force feedback. The results of this analysis (and in particular the contact instability phenomena) have been experimentally validated in [26], by using a simple test-bed, reproducing the case of a rigid link striking a rigid environment.

An alternative solution to possible bouncing and instability in impact control has been proposed in [23]: a positive acceleration feedback is inserted to control transient force response, to reduce the peak impulsive force and bouncing caused directly by overshooting and oscillation of transient force response. Considering the detection of the impact as an event, an event-driven switching control strategy is used to deal with the inadvertent loss of contact of the robot. The complete control scheme is then constituted by:

- a nonlinear feedback law, which linearises and decouples the robot dynamics;
- a switching controller, which governs the application of the proper control law (i.e. the position one or the force one);

- a PD position control law, which is applied in the free motion mode, during the transition phase, and in the constrained motion mode for the unconstrained directions only;
- a force control law, which is applied in the constrained directions during the contact phase, and constituted by a PI force law plus a weighted positive acceleration feedback term.

Such a control scheme, which has been validated both theoretically and experimentally on a 6-DOF PUMA 560 robot arm, has robustness properties with respect to different environments, but it requires the complete knowledge of the robot kinematics and dynamics, and the determination of the unconstrained and constrained directions during contact, so that its practical implementation may suffer for the same problems, which characterise the classical hybrid force/position control schemes.

3.3 A Unique Control Law for Free Motion and Contact

The main common, crucial characteristic of all the switching impact control schemes, which have been just illustrated, is given by the necessity to correctly detect impact (to switch to the proper controller), and then to handle the transition phase so to avoid rebounds, also in presence of possible delays in the control loop. The possibility of applying a unique control law for the free-motion phase and the contact one can then be quite appealing to avoid this kind of problems. In [19], such a solution has been discussed in the case of a redundant manipulator. Here, impact control is performed by reducing the impulsive forces at the end effector, both by means of a proper solution of robot redundancy and by the application of an impedance control scheme. In particular, redundancy is solved by following the *augmented kinematics approach*, by augmenting the Jacobian matrix on the basis of an impact model, which has been derived using the Cartesian space dynamic model of the manipulator. The resulting configuration of the redundant manipulator gives the smallest amount of impulsive force at the end effector, when it follows a prespecified Cartesian trajectory. The impact controller is based on a simplified impedance control scheme, in which the inverse of the desired inertia matrix is chosen to be identical to the mobility tensor of the manipulator in the Cartesian space, in order to reduce the impulsive forces, as well as the rebound effects. Only simulation results (no experimental ones) are provided for the validation of such a control scheme, so that it is not possible to evaluate its actual effectiveness in practice.

4 Some Proposed Impact Control Schemes

In this section, two different impact control strategies, which have been developed by the author and Tornambè in [12]–[16], are illustrated. Both solutions

utilise a unique controller both for the free motion phase and for the contact one, so that impact detection is not necessary: no force sensor is then introduced. It must be immediately stressed that the proposed control schemes do not correspond to “true” force controllers: such schemes simply provide (at least theoretically, and in the experimental tests that have been carried out) a good transition from the non-contact phase to the contact one, for all the robotic tasks in which the exact regulation of the contact force to a desired value is not required, while contact with unknown environments must be guaranteed, with the regulation of the robot to an assigned configuration.

The first control scheme, illustrated in Section 4.1, allows a stable contact of the robot with different environments (i.e. having different elastic/plastic characteristics) on the basis of joint positions and velocities measurements, without requiring the complete knowledge of the robot inertial parameters and of the environment stiffness.

The second control solution, illustrated in Section 4.2, is constituted by a position feedback control law, which has been developed under the assumption that velocity measures are not available, or that, if available, they may be incorrect at the first time instants after the impact, so that they cannot be used for control in the crucial transition phase from free motion to contact.

Both control schemes have been applied to a simple experimental test-bed, reproducing the case of a single-link robot striking rigid or elastic environments. The set-up, which is schematically reproduced in Figure 4.1, is briefly described hereafter.

The first steel bar, which is actuated by a DC motor, represents a single-link robot, while the environment that the link can strike is obtained by means of the second steel bar (not coaxial to the first one only for practical reasons), whose extremity, involved in the impact, can be: simply rested to the steel stroke end to reproduce the case of rigid impacts, covered with different materials to obtain quasi-rigid impacts, or connected to the stroke end by means of a spring to reproduce the conditions of a pure elastic impact.

Both bars have the same length, $l = 0.16$ m; the distance r_0 between their rotational axis (due to the physical dimensions of the motor, which could obstruct the positioning of the second bar) is of 0.052 m. The first bar (“the link”) is actuated by a brush type DC servo motor (Pittman, # GM9413D627), with voltage range of 12 V. Since the motor is a high-speed relatively low-torque actuator, it is geared down by a speed reducer, connecting the motor shaft to the link. The employed reducer, for which negligible backlash is ensured, has gear ratio of 127.7:1.

The output of the system is given by the link position, which is provided by an optical incremental encoder, mounted on the motor shaft and having a resolution of 400 cycles per revolution of the shaft. The position of the actuated link is then provided with a resolution of $400 \cdot 127.7$ cycles per revolution, corresponding to $7 \cdot 10^{-3}$ degrees (i.e. 0.123 mrad). As the encoder is of incremental type, a homing procedure is performed at the beginning of

each test to fix its zero position, with respect to which the link position will be measured during the test. If requested, the link angular velocity is computed on line via software from the measured position by a high-gain observer.

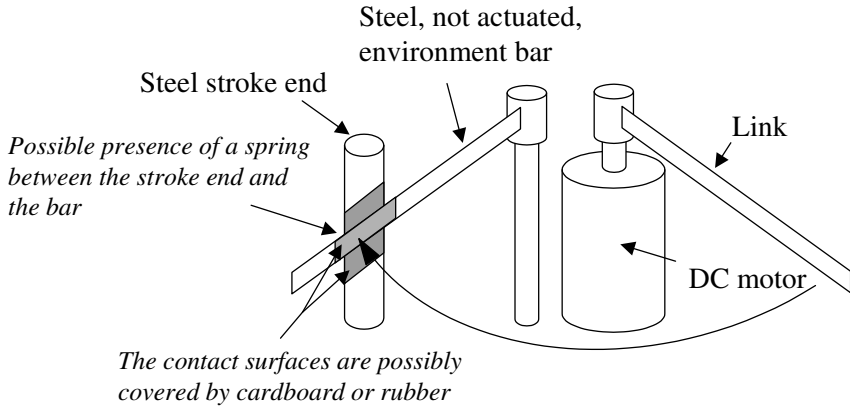


Figure 4.1. Sketch of the experimental test-bed

A second encoder is mounted on the unactuated bar (“the environment bar”) to detect the impact time, and to verify that contact between the link and the environment is never lost after the impact, *only to evaluate* the effectiveness of the proposed control schemes, but *not* for their implementation.

A unique power unit is used for the entire system, containing in particular: the servo amplifier to drive the DC motor and the encoders; the digital to analog converter for the determination of the command input signal to the motor; the counter for the software decoding of the encoders output. The system is interfaced for the control with a Pentium MMX 200 MHz Personal Computer by a Keithly digital to digital input/output board. The control and DSP software are both written in C++.

4.1 A Control Scheme Based on Impact Effects Estimation

A common approach, which utilises a unique control law for the two phases, is the basis of the impact control schemes developed in [12], [13], [15], and [24]. This kind of approach is based on the introduction of a reduced-order observer for the estimation of the impact induced forces, to be compensated

by the feedback control law. An interesting feature of this solution, which is illustrated hereafter directly with reference to the case reproduced by the experimental set-up, is given by its robustness properties. In fact, only “feasible” estimates of the robot dynamic parameters are sufficient for its implementation, whereas the knowledge of the environment stiffness properties is necessary only if a precise control force value must be imposed in the steady-state, but not to assure a stable contact.

Let $h(q)$ be the distance between the free extremity of the link and the contact surface of the environment bar, given by:

$$h(q(t)) = l \cos q(t) + r_0, \quad (4.1)$$

where $q(t)$ is the link angular position, r_0 is the distance between the rotational axis of the bars, having length l . The link is in contact with the environment bar when $h(q(t)) \leq 0$. The motion equations of the link can then be expressed as follows:

$$\ddot{q}(t) = \frac{1}{I}(u(t) - \beta \dot{q}(t)) + d_c(t), \quad (4.2)$$

where $I > 0$ is the link inertia, $\beta \geq 0$ is the viscous friction coefficient, $u(t)$ is the command input applied by the motor, and $d_c(t)$ is the impact-induced term, given by:

$$d_c(t) := \begin{cases} 0, & \text{if } h(q(t)) > 0 \\ -\frac{K}{I} J_h(q(t)) h(q(t)), & \text{if } h(q(t)) \leq 0, \end{cases} \quad (4.3)$$

where K is the unknown stiffness coefficient of the environment, and $J_h(q) := \frac{\partial h(q)}{\partial q} = -l \sin q$.

The control goal is bringing the link in a stable contact with the environment bar, while guaranteeing a proper desired value q_r for the steady-state position of the link, without using force measurements and/or detecting the impact time.

The proposed control scheme is constituted by a preliminary feedback control law:

$$u(t) = \hat{I} u'(t) + \hat{\beta} \dot{q}(t), \quad (4.4)$$

where $\hat{I} > 0$ and $\hat{\beta} \geq 0$ are some estimates of I and β , respectively, and by a second feedback law, given by:

$$u'(t) = -h_0(q(t) - q_r) - h_1 \dot{q}(t) - \hat{d}(t) \quad (4.5)$$

$$\hat{d}(t) = \xi(t) + \mu \dot{q}(t) \quad (4.6)$$

$$\dot{\xi}(t) = -\mu \xi(t) - \mu^2 \dot{q}(t) - \mu u'(t), \quad (4.7)$$

where h_0, h_1 are positive constants such that the roots of $p(\lambda) = \lambda^2 + h_1 \lambda + h_0$ are in the open left-half plane, and μ is a positive constant, which can be

chosen to obtain the global asymptotic stability of the closed-loop system (i.e. both in contact and non-contact conditions), as proven in [13].

The reduced-order observer, given by (4.6), (4.7), is used to evaluate both the impact-induced term $d_c(t)$ and the effect of the inexact compensation of the inertial and friction terms, performed by the preliminary feedback (4.4), by estimating the $d(t)$ term, given by:

$$d(t) := (\hat{I}_r - 1)u'(t) + \frac{\hat{I}_r}{\hat{I}} \Delta\beta\dot{q}(t) + d_c(t), \quad (4.8)$$

where $\hat{I}_r := \frac{\hat{I}}{I}$ and $\Delta\beta := \hat{\beta} - \beta$.

In order to guarantee the asymptotic stability of the closed-loop system, the observer gain μ must be greater than a threshold value, which depends on K , \hat{I}_r and $\Delta\beta$, and then on I and β (see [13] for details). In practice, it is not necessary to know them to implement the control scheme, as stability is achieved simply if μ is chosen sufficiently high. The weak point of the proposed control scheme is given by the fact that such a lower threshold value of μ increases as K does, so that more rigid contacts should require higher gain values, i.e. the controller could be inadequate when the environment is in practice highly rigid, and the control law is implemented on a computer with a small but not infinitesimal sampling time. Despite of this theoretical limitation, the experimental results that are illustrated hereafter (the interested reader can find more results in [15]) show that the proposed control law can be adopted also when the impacting parts are very rigid (both made of steel).

As no particular restriction has been made on the estimates \hat{I} and $\hat{\beta}$, the experimental tests have been carried out by considering $\hat{I} = 1$ and $\hat{\beta} = 0$, so that preliminary feedback (4.4) has not been actually performed, as it results in this case $u(t) = u'(t)$. The chosen values for the controller and observer gains are: $h_0 = 100$, $h_1 = 20$, $\mu = 0.1$; the sampling time is 0.005 s.

In the first carried out test, the environment bar is connected to the steel stroke end by means of a spring of unknown stiffness coefficient; the assigned reference link position, which is equal to 0.0942 rad, corresponds to a link rotation of 40 encoder impulses with respect to the zero position fixed by the initial homing procedure. The results reported in Figures 4.2 and 4.3 show that after the link impact against the environment bar, occurring after about 0.03 s (see Figure 4.3 (a)), the contact between the bars is never lost (see Figures 4.2 (a), 4.3 (a)), and the desired final position is actually reached by the link (see Figure 4.3 (b)).

No force measurement is available, but the $\hat{d}(t)$ term, evaluated by the observer (4.6), (4.7) and reported in Figure 4.2 (d), can be considered as an estimation of the effect of the impact induced forces on the command voltage during the contact phase. Only by the knowledge of the stiffness coefficient of the spring (not available in our case), the actual contact force value could

be computed from $\hat{d}(t)$, under the assumption that other disturbances (e.g. friction) are negligible during the contact phase.

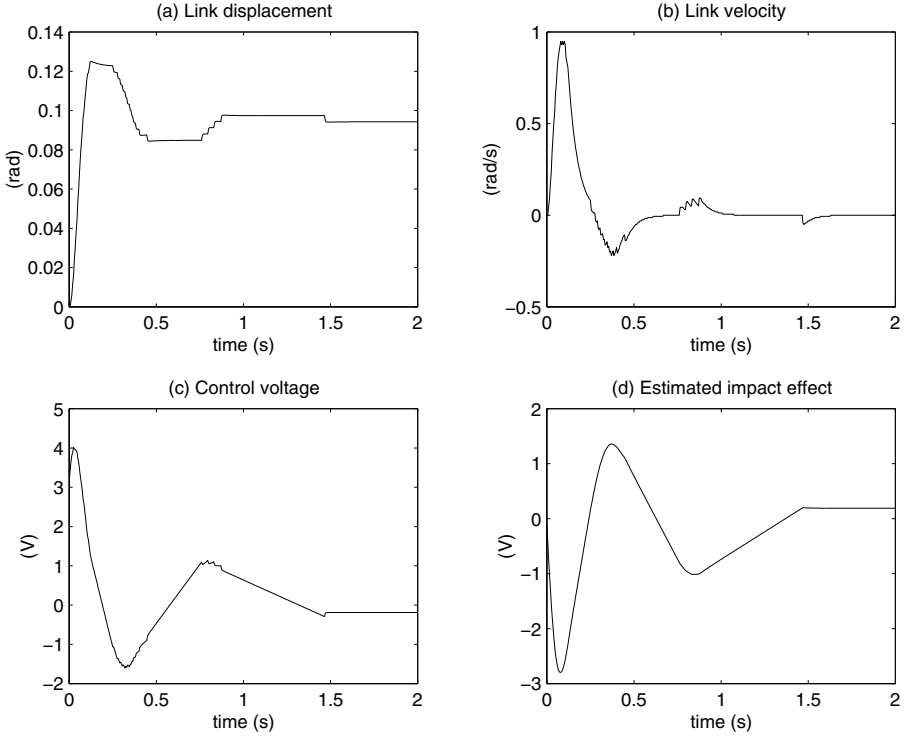


Figure 4.2. Experimental results: elastic impact, with the *controller + observer* scheme

From the theoretical point of view, the most critical situation for the proposed control scheme is represented by the case of a rigid impact. The results reported in Figures 4.4 and 4.5 are relative to the second carried out test, in which the environment bar is simply rested to the stroke end (without the insertion of any elastic element), and no material covers the steel bars: the impact can then be considered as rigid. The imposed reference value for the link position is now given by 0.1885 rad, defined with respect to the zero position fixed by the new initial homing procedure (note that the absence or presence of the spring changes the position of the environment bar); the controller and observer gains are kept unchanged.

As the results reported in Figures 4.4 and 4.5 show, also in this test the contact between the link and the environment bar, which is established after about 0.05 s, is indefinitely maintained (see in particular Figures 4.4 (a)

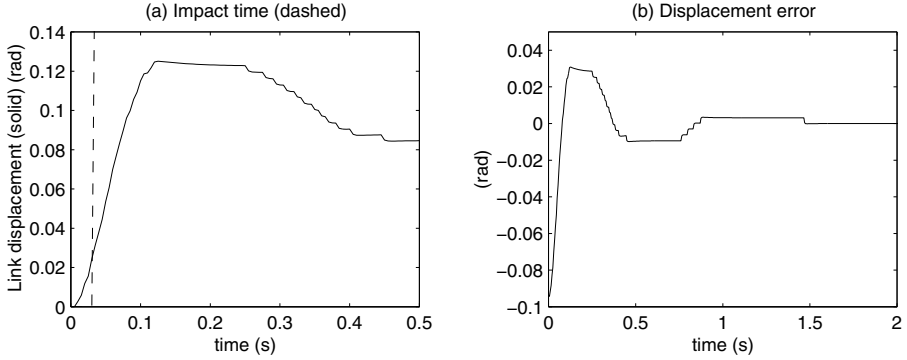


Figure 4.3. Experimental results: elastic impact, with the *controller + observer* scheme

and 4.5 (a)), and the desired link position is finally achieved, as shown by Figure 4.5 (b). As it was reasonable to expect, the effect of the impact induced force, given by the estimated term $\hat{d}(t)$, is larger than in the elastic case, especially in the first time instants after the impact (compare Figures 4.2 (d) and 4.4 (d)), but the corresponding control voltage is anyway largely tolerable by the system (see Figures 4.2 (c) and 4.4 (c)).

4.2 A Position Feedback Control Law

In papers [14], [16], a unique feedback control law has been proposed for simple mechanical systems subject to impacts (like the one represented by our experimental test-bed), by using only the output, given by the position of the actuated impacting element (the link in our case). Such a choice is motivated by the fact that the velocity measures, which are often available for the mechanical systems, can be incorrect at the first time instants, immediately after the impact.

For the sake of simplicity, the proposed control strategy is derived by considering a simplified representation of a mechanical system subject to impacts, and then directly extended to the single-link case of our experimental test-bed.

Consider a mass m , moving along a straight line, due to the action of a command force $f_c(t)$. The position of the mass is represented by $q(t)$ and measured with respect to a planar surface (located in $q = 0$), orthogonal to the motion line of the mass. The contact between the mass and the surface is established for all t such that $q(t) \leq 0$. The impact between the mass and the surface (described by a high stiffness coefficient K) is then defined as the transition from the mass free motion condition to the constrained motion one,

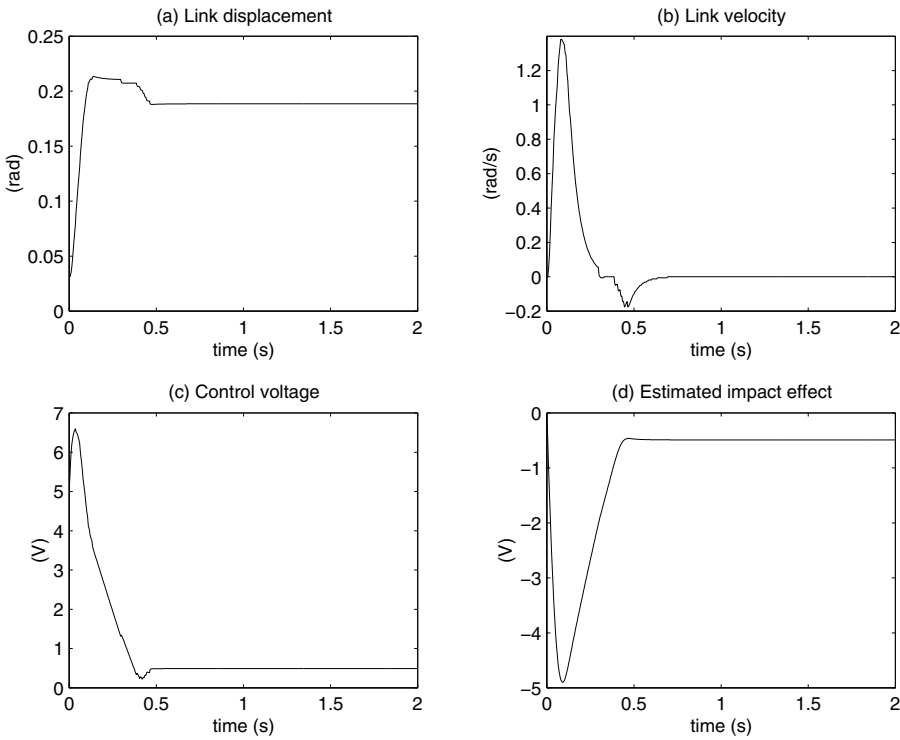


Figure 4.4. Experimental results: rigid impact, with the *controller + observer* scheme

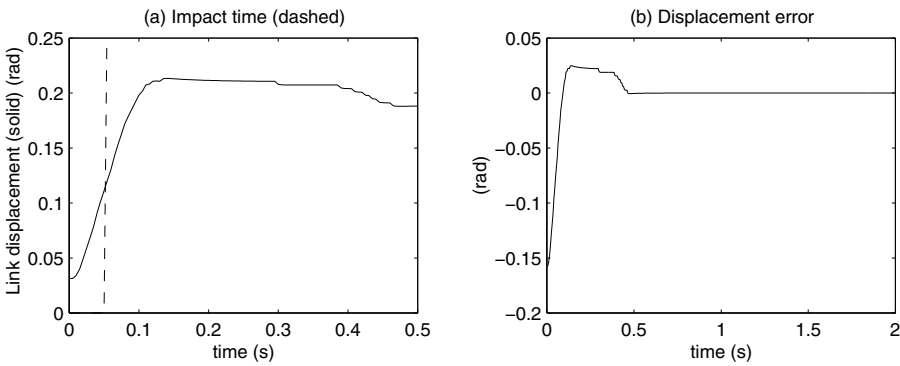


Figure 4.5. Experimental results: rigid impact, with the *controller + observer* scheme

at a nonzero negative velocity, i.e. impact occurs at a certain time instant t_i , if $q(t_i) = 0$ and $\dot{q}(t_i) < 0$.

By considering $y(t) = q(t)$ as output, and $u(t) = \frac{1}{m}f_c(t)$ as input, the following transfer functions $P_i(s) := Y(s)/U(s)$, $i = 1, 2$, can be defined in the two non-contact/contact situations:

$$P_1(s) = \frac{1}{s^2} \quad (\text{non-contact}) \quad P_2(s) = \frac{1}{s^2 + k} \quad (\text{contact}), \quad (4.9)$$

where $k := K/m$.

The control goal, to be performed by developing a unique position feedback law, whose structure is independent of the motion phase (contact or non-contact), is to bring the mass in a stable contact with the surface, while achieving the practical regulation of the mass position to an assigned reference value (corresponding to a desired contact force, if K is known), and guaranteeing a satisfactory insensitivity of the system behaviour to the impact effects, i.e. with a small force overshoot due to the impacts.

As for the control scheme illustrated in the previous subsection, the control goal is expressed with reference to a desired “contact” position, since no force sensor is assumed to be used; besides, in this case, as no estimation of the impact induced forces or effects is performed, the goal is intended to be completely accomplished if the controlled system automatically reduces the impact effects in a satisfying way for the requested task, and for the preservation of the mechanical system itself.

The first step to be performed for the design of the control feedback law is checking if systems $P_1(s)$ and $P_2(s)$ are simultaneously stabilisable. By introducing coprime rational factorisations of $P_1(s)$ and $P_2(s)$ of the form $P_i(s) = N_{P_i}(s)/D_{P_i}(s)$, $i = 1, 2$, where $N_{P_i}(s), D_{P_i}(s) \in \mathcal{R}$ (with \mathcal{R} being the set of the proper rational functions, having all the poles in the open left-half plane), it is possible to prove [4], [5] that there exists a one-parameter family of compensators $G(s)$, such that the closed-loop system corresponding to the open-loop transfer function $G(s)P_1(s)$ in the non-contact case, and to $G(s)P_2(s)$ in the contact one, is internally asymptotically stable in both phases, i.e. systems $P_1(s)$ and $P_2(s)$ can be simultaneously stabilised (see details in [16]). It follows then that the global closed-loop system thus obtained is internally asymptotically stable if a finite number of impacts occurs, i.e. if a finite number of switches between the transfer functions in (4.9) occurs. If it is not possible to prove that the number of impacts is finite, the asymptotic stability of the mechanical system in the two separate phases (the non-contact and the contact one) *does not* imply the asymptotic stability of the global system, as it is strongly dependent on how the system commutes between the two different phases, and a deeper investigation is then needed.

For the subsequent developments, it is useful to define the term:

$$\delta(t) = \begin{cases} 0 & \text{if } q(t) > 0 \text{ (non-contact)} \\ kq(t) & \text{if } q(t) \leq 0 \text{ (contact)}, \end{cases} \quad (4.10)$$

which is proportional to the impact induced force.

By inserting the control law:

$$u(s) = G(s)(v(s) - y(s)), \quad (4.11)$$

where $v(s)$ is the Laplace transform of the reference signal $v(t)$, and $G(s)$ is the transfer function of the compensator to be determined, the following closed-loop system is obtained:

$$e(s) = \frac{1}{1 + P_1(s)G(s)}v(s) + \frac{P_1(s)}{1 + P_1(s)G(s)}\delta(s), \quad (4.12)$$

where $e(s)$ is the Laplace transform of the tracking error $e(t) := v(t) - y(t)$.

To accomplish the previously described control goal, $G(s)$ must be determined so that:

- a) the closed-loop system is asymptotically stable (independently of the number of impacts);
- b) $W_e(s) := \frac{1}{1 + P_1(s)G(s)}$ has at least a zero in $s = 0$, to guarantee that the tracking error is null at the steady-state, for any constant reference input $v(t) = \bar{v}$;
- c) $W_d(s) := \frac{P_1(s)}{1 + P_1(s)G(s)}$ has at least a zero in $s = 0$ too, for the tracking error to be equal to zero at the steady-state, also in presence of nonzero steady-state values of the contact force, and it is such that the impact effects on the measured output are reduced as much as possible during the transient time (in the ideal case it would be desirable to have no effect at all).

It can be proven [5] that the set of all the compensators $G(s)$ that stabilises $P_1(s)$ can be parameterised as follows:

$$G(s) = \frac{A_1(s) + Q(s)C_1(s)}{B_1(s) + Q(s)D_1(s)}, \quad (4.13)$$

where $A_1(s)$, $B_1(s)$, $C_1(s)$, and $D_1(s)$ are defined by the previously introduced coprime rational factorisation of $P_1(s)$, according to the Bezout identity [5], and $Q(s) \in \mathcal{R}$ is such that $L_1(s) = B_1(s) + Q(s)D_1(s)$ is biproper (as $P_1(s)$ is strictly proper, $L_1(s)$ is surely biproper for all $Q(s) \in \mathcal{R}$).

The compensator $G(s)$ given in (4.13) certainly stabilises $P_1(s)$ and (it can be easily verified) satisfies property b), whereas the *ideal* choice for $Q(s)$, to fully satisfy the properties requested by c), would be given by a not proper, i.e. not implementable, function $Q_{id}(s)$ of the form: $Q_{id}(s) = -(k_0s^2 + k_1s + k_2)$, where k_i , $i = 0, 1, 2$ are determined by the used factorisation of $P_1(s)$. $Q(s)$ can then be defined so as to approximate the ideal function $Q_{id}(s)$ at least in the operating frequency bandwidth $[0, \Omega]$, where Ω is a sufficiently high frequency; a possible choice for $Q(s)$ is given by:

$$Q(s) = -\frac{(k_0 s^2 + k_1 s + k_2)}{(1 + \tau_1 s)(1 + \tau_2 s)}, \quad (4.14)$$

where the time constants τ_1, τ_2 of the poles must be such that $\frac{1}{\tau_1}, \frac{1}{\tau_2} \gg \Omega$.

As it is not possible to know if a finite number of impacts occurs (i.e. a finite number of switches between $P_1(s)$ and $P_2(s)$), a sufficient condition for the asymptotic stability of the entire closed-loop system can be found by determining a unique Lyapunov function for the closed-loop system in the two phases, once a state-space realisation has been introduced for $G(s)$ (see details and simulation results in [16]).

Th effectiveness of such a control solution has been validated by using our experimental test-bed in the case of rigid impacts (i.e. with the environment bar simply rested to the stroke end). The command input has been defined from (4.13) in the following form:

$$u(s) = K_P(\bar{v} - y(s)) - sK_D y(s) + \alpha y_Q(s), \quad (4.15)$$

in which $y_Q(s)$ is the output of the system that realises the transfer function $Q(s)$ given in (4.14), whereas the value of the coefficient α depends on the link inertia.

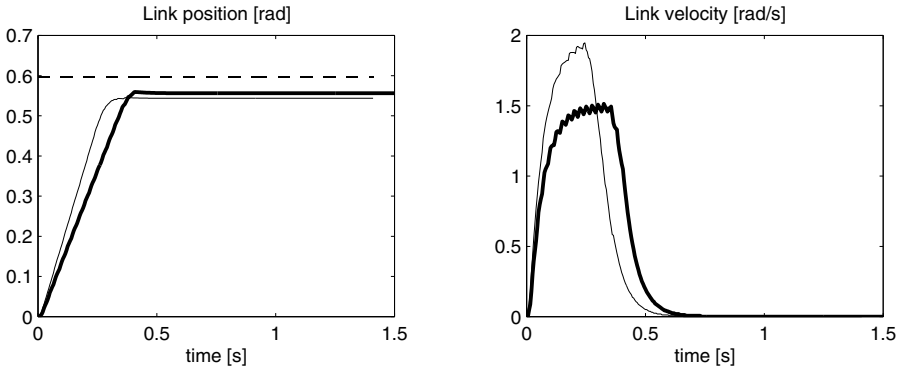


Figure 4.6. Experimental results: rigid impact, with the *position feedback* control scheme (with $K_P = 100$, $K_D = 20$, $\tau_1 = \tau_2 = 0.5$ s)

The sampling time used for the discrete-time implementation of the controller is 0.005 s, as in the previous tests, whereas the discretisation has been performed by using the first order Euler approximation rule.

A link rotation of 190 encoder impulses, corresponding to a reference position value (compatible with the contact of the link with the environment bar) of 0.5969 rad, is imposed, by considering $K_P = 100$, $K_D = 20$ in (4.15).

Figure 4.6 shows the obtained time-history of the link position and velocity without the insertion of the $y_Q(s)$ term in (4.15) (reported by a thin line), and with the insertion of $y_Q(s)$ (reported by a bold line), corresponding to the output of

$$Q(s) = -\frac{4s^2 + 86s + 523}{s^2 + 4s + 4},$$

i.e. with $\tau_1 = \tau_2 = 0.5$ s.

As expected by the previously developed analysis, the link is led towards the environment bar, and a stable contact is established between them. The insertion of the y_Q term allows a significant reduction of the impact effects, corresponding to a reduction of the link maximum velocity of about 23%, and also a reduction of the steady-state position tracking error (the desired position value is reported in the figure by a dashed line). No force time-history can be reported, as no force sensor has been employed, and the stiffness properties of the environment bar are not exactly known.

To evaluate the influence of the y_Q term on the reduction of the link maximum velocity, the same reference position has been considered in a new test, in which the choices $K_P = 300$ and $K_D = 5$ have been made in (4.15) (i.e. a very small damping action has been introduced in the PD part of the command law). The choice $\tau_1 = \tau_2 = 0.5$ s has been maintained for $Q(s)$. Figure 4.7 shows the obtained time-history of the link position and velocity without the insertion of the $y_Q(s)$ term in (4.15) (reported by a thin line), and with the insertion of $y_Q(s)$ (reported by a bold line).

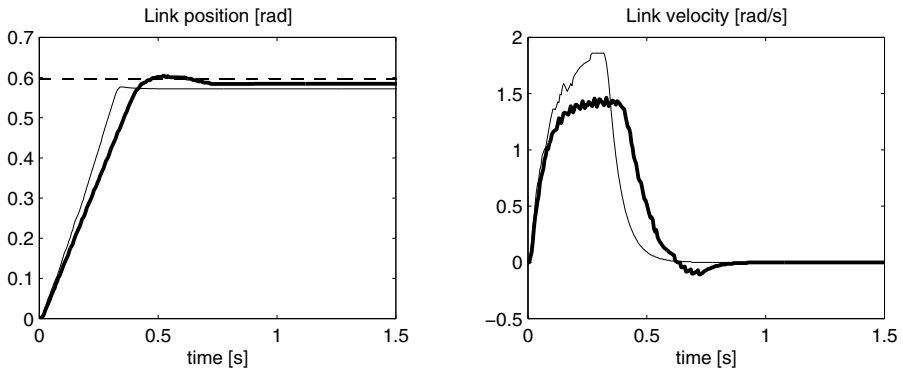


Figure 4.7. Experimental results: rigid impact, with the *position feedback* control scheme (with $K_P = 300$, $K_D = 5$, $\tau_1 = \tau_2 = 0.5$ s)

As the figure shows, also in this case a stable contact between the bars is achieved, together with a significant reduction of the impact effects when the y_Q term is inserted. In particular, a reduction of the link maximum velocity

of about 25% is obtained, and a reduced tracking error is achieved, with only a negligible position overshoot with respect to the assigned reference value (reported in the figure by a dashed line).

The obtained results have shown then the effectiveness of the proposed approach, despite of all the factors that have been neglected in the developed analysis (such as any form of friction and stiction, the coefficient of restitution, etc.). In particular, they have shown that, even if the exact achievement of the desired link position can be prevented in practice, mainly due to friction, maintenance of a stable contact is assured with a satisfying reduction of the impact effects.

5 Conclusions

Interesting and quite good results have been obtained during the last years in the impact control field, but the problem is still quite open. Performance of the presented schemes can depend on various elements, such as the quality of the used sensors and actuators, the actual knowledge of the systems involved in the impact, and the impact type itself, so that the choice of the *best* strategy for each application is often a delicate task, that the designer must afford taking into account all the technological characteristics and limitations of the considered plant.

Finally, a further deeper analysis should be made for the micro-robotics applications and for the robots having flexible links, since the reduced dimensions and the distributed elasticity properties of the involved structures should probably imply a more detailed description of the impact phenomena, for the subsequent design of suitable control strategies.

Acknowledgements

The author wishes to thank Antonio Tornambè (coauthor of papers [12]–[16]), with whom the research about impact control has been carried out.

The research activity of the Turin Polytechnic group has been cofunded by *Italian Space Agency* (ASI) under grant ARS98-163.

References

1. Brogliato B 1999 *Nonsmooth Mechanics: Models, Dynamics, Control* (2nd ed) Springer-Verlag, London, UK
2. Brogliato B, Niculescu S-I, Orhant P 1997 On the control of finite-dimensional mechanical systems with unilateral constraints. *IEEE Transactions on Automatic Control* 42:200–215

3. Chapnik B, Heppler G, Aplevich J 1991 Modelling impact on a one-link flexible robotic arm. *IEEE Transactions on Robotics and Automation* 7:479–488
4. Dorato P, Fortuna L, Muscato G 1992 *Robust control for unstructured perturbations: an introduction* Springer-Verlag, Berlin, D
5. Doyle J C, Francis B A, Tannenbaum A R 1992 *Feedback Control Theory* Macmillan, New York, NY
6. Ferretti G, Magnani G, Zavala Rio A 1998 Impact modeling and control for industrial manipulators. *IEEE Control Systems Magazine* 18(4):65–71
7. Goldsmith W 1960 *Impact* Edward Arnold Publisher, London, UK
8. Hogan N 1985 Impedance control: An approach to manipulation: Part I - Theory, Part II - Implementation, and Part II - Applications. *ASME Journal of Dynamic Systems, Measurements and Control* 107:1–24
9. Hurmuzlu Y, Chang T H 1992 Rigid body collisions of a special class of planar kinematic chains. *IEEE Transactions on Systems, Man and Cybernetics* 22:964–971
10. Hurmuzlu Y, Marghitu D B 1994 Rigid body collision of planar kinematic chains with multiple contact points. *International Journal of Robotics Research* 13:82–92
11. Indri M 2000 Impact control in robotics: A brief survey. In: *Preprints of 6th IFAC Symposium on Robot Control* Vienna, A, pp 97–102
12. Indri M, Tornambè A 1999 Impact model and control of two multi-dof cooperating manipulators. *IEEE Transactions on Automatic Control* 44:1297–1303
13. Indri M, Tornambè A 2000 Robust control of a simple mechanical system subject to impacts. In: *Preprints of 3rd IFAC Symposium on Robust Control Design* Prague, CZ
14. Indri M, Tornambè A 2000 Design of an output feedback control law for a mechanical system subject to impacts: Output regulation and impact insensitivity. In: *Proceedings of 2nd International Symposium on Impact and Friction of Solids, Structures and Machines* Montréal, CAN
15. Indri M, Tornambè A 2000 Impact control of a single-link robot striking different environments: Theoretical and experimental investigation. *European Journal of Control* 6(4)
16. Indri M, Tornambè A 2001 On the design of a position feedback control law for a simple mechanical system subject to impacts. *International Journal of Control*, 74
17. Johnson W 1972 *Impact Strength of Materials* Crane, Russak and Co, New York, NY
18. Kazerooni H 1991 Robust nonlinear impedance control for robot manipulators. In: *Proceedings of 1991 IEEE International Conference on Robotics and Automation* Sacramento, CA, pp 741–750
19. Lin Z C, Patel R V, Balafoutis C A 1995 Impact reduction for redundant manipulators using augmented impedance control. *Journal of Robotic Systems* 12:301–313
20. Mandal N, Payandeh S 1995 Control strategies for robotic contact tasks: An experimental study. *Journal of Robotic Systems* 12:67–92
21. Mills J K, Lokhorst D M 1993 Control of robotic manipulators during general task execution: A discontinuous control approach. *International Journal of Robotics Research* 12:146–163
22. Niculescu S-I, Brogliato B 1999 Force measurement time-delays and contact instability phenomenon. *European Journal of Control* 5:275–289
23. Tarn T-J, Wu Y, Xi N, Isidori A 1996 Force regulation and contact transition control. *IEEE Control Systems Magazine* 16(1):32–40

24. Tornambè A 1996 Global regulation of a planar robot arm striking a surface. *IEEE Transactions on Automatic Control* 41:1517–1521
25. Tornambè A 1999 Modeling and control of impact in mechanical systems: Theory and experimental results. *IEEE Transactions on Automatic Control* 44:294–309
26. Tornambè A 1999 Discussion on “Force measurement time-delays and contact instability phenomenon”, by SI Niculescu and B Brogliato. *European Journal of Control* 5:290–292
27. Volpe R, Khosla P 1993 A theoretical and experimental investigation of impact control for manipulators. *International Journal of Robotics Research* 13:351–365
28. Volpe R, Khosla P 1993 A theoretical and experimental investigation of explicit torque control strategies for manipulators. *IEEE Transactions on Automatic Control* 38:1634–1649
29. Youcef-Toumi K, Gutz D A 1994 Impact and force control: modelling and experiments. *ASME Journal of Dynamic Systems, Measurements and Control* 116:89–98
30. Zukas J A 1982 *Impact Dynamics* Wiley, New York, NY

Control of Wheeled Mobile Robots: An Experimental Overview

Alessandro De Luca, Giuseppe Oriolo, and Marilena Vendittelli

Dipartimento di Informatica e Sistemistica
Università degli Studi di Roma “La Sapienza”
<http://labrob.ing.uniroma1.it>

The subject of this chapter is the motion control problem of wheeled mobile robots (WMRs). With reference to the unicycle kinematics, we review and compare several control strategies for trajectory tracking and posture stabilization in an environment free of obstacles. Experiments are reported for SuperMARIO, a two-wheel differentially-driven mobile robot. From the comparison of the obtained results, guidelines are provided for WMR end-users.

1 Introduction

Wheeled mobile robots (WMRs) are increasingly present in industrial and service robotics, particularly when flexible motion capabilities are required on reasonably smooth grounds and surfaces [29]. Several mobility configurations (wheel number and type, their location and actuation, single- or multi-body vehicle structure) can be found in the applications, e.g. see [18]. The most common for single-body robots are differential drive and synchro drive (both kinematically equivalent to a unicycle), tricycle or car-like drive, and omnidirectional steering. A detailed reference on the analytical study of the kinematics of WMRs is [2].

Beyond the relevance in applications, the problem of autonomous motion planning and control of WMRs has attracted the interest of researchers in view of its theoretical challenges. In particular, these systems are a typical example of nonholonomic mechanisms due to the perfect rolling constraints on the wheel motion (no longitudinal or lateral slipping) [24].

In the absence of workspace obstacles, the basic motion tasks assigned to a WMR may be reduced to moving between two robot postures and following a given trajectory. From a control viewpoint, the peculiar nature of nonholonomic kinematics makes the second problem easier than the first; in fact, it is known [8] that feedback stabilization at a given posture cannot be achieved via smooth time-invariant control. This indicates that the problem is truly nonlinear; linear control is ineffective, even locally, and innovative design techniques are needed.

After a preliminary attempt at designing local controllers, the trajectory tracking problem was globally solved in [28] by using a nonlinear feedback

action, and independently in [12] and [11] through the use of dynamic feedback linearisation. A recursive technique for trajectory tracking of nonholonomic systems in chained form can also be derived from the backstepping paradigm [17]. As for posture stabilization, both discontinuous and/or time-varying feedback controllers have been proposed. Smooth time-varying stabilization was pioneered by Samson [26], [27], while discontinuous (often, time-varying) control was used in various forms, e.g. see [1], [10], [21], [22], [32]. A recent addition to this class was presented in [14], where dynamic feedback linearisation has been extended to the posture stabilization problem.

While comparative simulations of several of the above methods are given in [9] for a unicycle and in [13] for a car-like vehicle, there is no extensive experimental testing on a single benchmark vehicle. The objective of this chapter is therefore to evaluate and compare the practical design and performance of control methods for trajectory tracking and posture stabilization, highlighting potential implementation problems related to kinematic or dynamic nonidealities, e.g. wheel slippage, discretisation and quantization of signals, friction and backlash, actuator saturation and dynamics.

All control designs are directly presented for the case of unicycle kinematics, the most common among WMRs, and experimentally tested on the laboratory prototype SuperMARIO. Nonetheless, most of the methods selected for comparison can be generalised to vehicles with more complex kinematics.

1.1 Organization of Contents

In Section 2 we classify the basic motion control tasks for WMRs. The modeling and main control properties are summarised in Section 3. In Section 4, the experimental setup used in our tests is described in detail.

Trajectory tracking controllers are presented in Section 5. After discussing the role of nominal feedforward commands (Section 5.1), three feedback laws are illustrated. They are based respectively on tangent linearisation along the reference trajectory and linear control design (Section 5.2), on a nonlinear Lyapunov-based control technique (Section 5.3), and on the use of dynamic feedback linearisation (Section 5.4). Comparative experiments on exact and asymptotic trajectory tracking are conducted in Section 5.5, using an eight-shaped desired trajectory.

The posture stabilization problem to the origin of the configuration space is considered in Section 6. Four conceptually different feedback methods are presented, using time-varying smooth (Section 6.1) or nonsmooth (Section 6.2) control laws, a discontinuous controller based on polar coordinates transformation (Section 6.3), and a stabilizing law based on dynamic feedback linearisation (Section 6.4). Results on forward and parallel parking experiments are reported.

Finally, in Section 7 the obtained results are summarised and compared in terms of performance, ease of control parameters tuning, sensitivity to nonidealities, and generalisability to other WMRs. In this way, some guidelines

are proposed to end-users interested in implementing control laws for WRMs. Open problems for further research are pointed out.

2 Basic Motion Tasks

The basic motion tasks that we consider for a WMR in an obstacle-free environment are (see Figure 2.1):

- *Point-to-point motion*: The robot must reach a desired goal configuration starting from a given initial configuration.
- *Trajectory following*: A reference point on the robot must follow a trajectory in the Cartesian space (i.e. a geometric path with an associated timing law) starting from a given initial configuration.

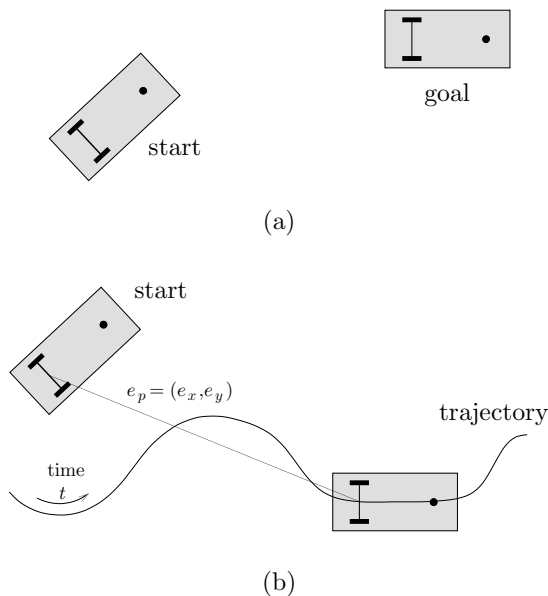


Figure 2.1. Basic motion tasks for a WMR: (a) point-to-point motion; (b) trajectory following

Execution of these tasks can be achieved using either feedforward commands, or feedback control, or a combination of the two. Indeed, feedback solutions exhibit an intrinsic degree of robustness. However, especially in the case of point-to-point motion, the design of feedback laws for nonholonomic systems has to face a serious structural obstruction, as we will show in Section 3; controllers that overcome such difficulty may lead to unsatisfactory transient

performance. The design of feedforward commands is instead strictly related to trajectory planning, whose solution should take into account the specific nonholonomic nature of the WMR kinematics.

When using a feedback strategy, the point-to-point motion task leads to a *state regulation* control problem for a point in the robot state space — *posture stabilization* is another frequently used term. Without loss of generality, the goal can be taken as the origin of the n -dimensional robot configuration space. As for trajectory following, in the presence of an initial error (i.e. an off-trajectory start for the vehicle) the *asymptotic tracking* control problem consists in the stabilization to zero of $e_p = (e_x, e_y)$, the two-dimensional Cartesian error with respect to the position of a moving reference robot (see Figure 2.1b).

Contrary to the usual situation, tracking is easier than regulation for a nonholonomic WMR. An intuitive explanation of this can be given in terms of a comparison between the number of controlled variables (outputs) and the number of control inputs. For the unicycle-like vehicle of Section 3, two input commands are available while three variables (x , y , and the orientation θ) are needed to determine its configuration. Thus, regulation of the WMR posture to a desired configuration implies zeroing three independent configuration errors. When tracking a trajectory, instead, the output e_p has the same dimension as the input and the control problem is square.

3 Modelling and Control Properties

Let $q \in \mathcal{Q}$ be the n -vector of generalised coordinates for a wheeled mobile robot. *Pfaffian* nonholonomic systems are characterised by the presence of $n - m$ non-integrable differential constraints on the generalised velocities of the form

$$A(q)\dot{q} = 0. \quad (3.1)$$

For a WMR, these arise from the rolling without slipping condition for the wheels. All feasible instantaneous motions can then be generated as

$$\dot{q} = G(q)w, \quad w \in \mathbb{R}^m, \quad (3.2)$$

where the columns g_i , $i = 1, \dots, m$, of the $n \times m$ matrix $G(q)$ are chosen so as to span the null space of matrix $A(q)$. Different choices are possible for G , according to the physical interpretation that can be given to the ‘weights’ w_1, \dots, w_m . Equation (3.2), which is called the (first-order) *kinematic model* of the system, represents a driftless nonlinear system.

The simplest model of a nonholonomic WMR is that of the *unicycle*, which corresponds to a single upright wheel rolling on the plane (top view in Figure 3.1). The generalised coordinates are $q = (x, y, \theta) \in \mathcal{Q} = \mathbb{R}^2 \times SO^1$ ($n = 3$). The constraint that the wheel cannot slip in the lateral direction is given in the form (3.1) as

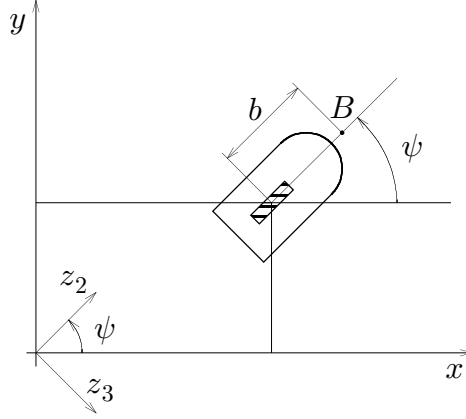


Figure 3.1. Relevant variables for the unicycle (top view)

$$\dot{x} \sin \theta - \dot{y} \cos \theta = 0.$$

A kinematic model is thus

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = g_1(q)v + g_2(q)\omega = \begin{bmatrix} \cos \theta \\ \sin \theta \\ 0 \end{bmatrix} v + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \omega, \quad (3.3)$$

where v and ω (respectively, the linear velocity of the wheel and its angular velocity around the vertical axis) are assumed as available control inputs ($m = 2$). As we will show in Section 4, this model is equivalent to that of SuperMARIO.

System (3.3) displays a number of structural control properties, most of which actually hold more in general for (3.2).

3.1 Controllability at a Point

The tangent linearisation of (3.3) at any point q_e is the linear system

$$\dot{\tilde{q}} = \begin{bmatrix} \cos \theta_e \\ \sin \theta_e \\ 0 \end{bmatrix} v + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \omega \quad \tilde{q} = q - q_e,$$

that is clearly not controllable. This implies that a linear controller will never achieve posture stabilization, not even in a local sense. In order to study the controllability of the unicycle, we need therefore to use tools from nonlinear control theory [16]. It is easy to check that the *accessibility rank condition* is satisfied globally (at any q_e), since

$$\text{rank} \begin{bmatrix} g_1 & g_2 & [g_1, g_2] \end{bmatrix} = 3 = n, \quad (3.4)$$

being the Lie bracket $[g_1, g_2]$ of the two input vector fields

$$[g_1, g_2] = \frac{\partial g_2}{\partial q} g_1 - \frac{\partial g_1}{\partial q} g_2 = \begin{bmatrix} \sin \theta \\ -\cos \theta \\ 0 \end{bmatrix}.$$

Since the system is driftless, condition (3.4) implies its controllability.

Controllability can also be shown constructively, i.e. by providing an explicit sequence of maneuvers bringing the robot from any start configuration (x_s, y_s, θ_s) to any desired goal configuration (x_g, y_g, θ_g) . Since the unicycle can rotate on itself, this task is simply achieved by an initial rotation on (x_s, y_s) until the unicycle is oriented toward (x_g, y_g) , followed by a translation to the goal position, and by a final rotation on (x_g, y_g) so as to align θ with θ_g .

As for the stabilisability of system (3.3) to a point, the failure of the previous linear analysis indicates that exponential stability cannot be achieved by smooth feedback [31]. Things turn out to be even worse: if smooth (in fact, even continuous) time-invariant feedback laws are used, Lyapunov stability is out of reach. This negative result is established on the basis of a necessary condition due to Brockett [6]: smooth stabilisability of a driftless *regular* system (i.e. such that the input vector fields are well defined and linearly independent at q_e) requires a number of inputs equal to the number of states.

The above obstruction has a deep impact on the control design. In fact, to obtain a posture stabilizing controller it is either necessary to give up the continuity requirement and/or to resort to time-varying control laws. In Section 6 we shall pursue both approaches.

3.2 Controllability About a Trajectory

Given a desired Cartesian motion for the unicycle, it may be convenient to generate a corresponding *state* trajectory $q_d(t) = (x_d(t), y_d(t), \theta_d(t))$. In order to be feasible, the latter must satisfy the nonholonomic constraint on the vehicle motion or, equivalently, be consistent with (3.3). The generation of $q_d(t)$ and of the corresponding reference velocity inputs $v_d(t)$ and $\omega_d(t)$ will be addressed in Section 5.

Defining the state tracking error as $\tilde{q} = q - q_d$ and the input variations as $\tilde{v} = v - v_d$ and $\tilde{\omega} = \omega - \omega_d$, the tangent linearisation of system (3.3) about the reference trajectory is

$$\dot{\tilde{q}} = \begin{bmatrix} 0 & 0 & -v_d \sin \theta_d \\ 0 & 0 & v_d \cos \theta_d \\ 0 & 0 & 0 \end{bmatrix} \tilde{q} + \begin{bmatrix} \cos \theta_d & 0 \\ \sin \theta_d & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{v} \\ \tilde{\omega} \end{bmatrix} = A(t)\tilde{q} + B(t) \begin{bmatrix} \tilde{v} \\ \tilde{\omega} \end{bmatrix}. \quad (3.5)$$

Since the linearised system is *time-varying*, a necessary and sufficient controllability condition is that the controllability Gramian is nonsingular. However, a simpler analysis can be conducted by defining the state tracking error through a rotation matrix as

$$\tilde{q}_R = \begin{bmatrix} \cos \theta_d & \sin \theta_d & 0 \\ -\sin \theta_d & \cos \theta_d & 0 \\ 0 & 0 & 1 \end{bmatrix} \tilde{q}.$$

Using (3.5), we obtain

$$\dot{\tilde{q}}_R = \begin{bmatrix} 0 & \omega_d & 0 \\ -\omega_d & 0 & v_d \\ 0 & 0 & 0 \end{bmatrix} \tilde{q}_R + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{v} \\ \tilde{\omega} \end{bmatrix}. \quad (3.6)$$

When v_d and ω_d are constant, the above linear system becomes time-invariant and controllable, since matrix

$$C = [B \quad AB \quad A^2B] = \begin{bmatrix} 1 & 0 & 0 & 0 & -\omega_d^2 & v_d\omega_d \\ 0 & 0 & -\omega_d & v_d & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

has rank 3 provided that either v_d or ω_d are nonzero. Therefore, we conclude that the kinematic system (3.3) can be locally stabilised by linear feedback about trajectories which consist of linear or circular paths, executed with constant velocity.

In Section 5 we shall see that it is possible to use linear design techniques in order to obtain local stabilization for arbitrary feasible trajectories, provided they do not come to a stop.

3.3 Feedback Linearisability

Based on the previous discussion, it is easy to see that the driftless nonholonomic system (3.2) cannot be transformed into a linear controllable one using *static* state feedback. In particular, for the unicycle (3.3) the controllability condition (3.4) implies that the distribution generated by vector fields g_1 and g_2 is not involutive, thus violating the necessary condition for full state feedback linearisability [16].

However, when matrix $G(q)$ in (3.2) has full column rank, m equations can always be transformed via feedback into simple integrators (input-output linearisation and decoupling). The choice of the linearising outputs is not unique and can be accommodated for special purposes. An interesting example is the following. Define the two outputs as

$$\begin{aligned} y_1 &= x + b \cos \theta \\ y_2 &= y + b \sin \theta, \end{aligned}$$

with $b \neq 0$, i.e. the Cartesian coordinates of a point B displaced at a distance b along the main axis of the unicycle (see Figure 3.1).

Using the globally defined state feedback

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta/b & \cos \theta/b \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix},$$

the unicycle is equivalent to

$$\begin{aligned} \dot{y}_1 &= u_1 \\ \dot{y}_2 &= u_2 \\ \dot{\theta} &= \frac{u_2 \cos \theta - u_1 \sin \theta}{b}. \end{aligned}$$

As a consequence, a linear feedback controller for $u = (u_1, u_2)$ will make the point B track *any* reference trajectory, even with discontinuous tangent to the path (e.g. a square without stopping at corners). Moreover, it is easy to show that the internal state evolution $\theta(t)$ is bounded. This approach, however, will not be pursued in this chapter because of its limited interest for more general kinematics.

For exact linearisation purposes, one may also resort to the more general class of *dynamic* state feedback. In this case, the conditions for full state linearisation are less stringent and turn out to be satisfied for a large class of nonholonomic WMRs. However, there is a potential control singularity that has to be considered carefully. The use of dynamic feedback linearisation will be illustrated later both for asymptotic trajectory tracking (Section 5) and for posture stabilization (Section 6).

3.4 Chained Forms

The existence of canonical forms for kinematic models of nonholonomic robots allows a general and systematic development of both open-loop and closed-loop control strategies. The most useful canonical structure is the *chained form*, which in the case of two-input systems is

$$\begin{aligned} \dot{z}_1 &= u_1 \\ \dot{z}_2 &= u_2 \\ \dot{z}_3 &= z_2 u_1 \\ &\vdots \\ \dot{z}_n &= z_{n-1} u_1. \end{aligned} \tag{3.7}$$

It has been shown that a two-input driftless nonholonomic system with up to $n = 4$ generalised coordinates can always be transformed in chained form by static feedback transformation [23]. As a matter of fact, most (but not all) WMRs can be transformed in chained form.

For the kinematic model (3.3) of the unicycle, we introduce the following globally defined coordinate transformation

$$\begin{aligned}
z_1 &= \theta \\
z_2 &= x \cos \theta + y \sin \theta \\
z_3 &= x \sin \theta - y \cos \theta
\end{aligned}$$

and static state feedback

$$\begin{aligned}
v &= u_2 + z_3 u_1 \\
\omega &= u_1,
\end{aligned} \tag{3.8}$$

obtaining

$$\begin{aligned}
\dot{z}_1 &= u_1 \\
\dot{z}_2 &= u_2 \\
\dot{z}_3 &= z_2 u_1.
\end{aligned} \tag{3.9}$$

Note that (z_2, z_3) is the position of the unicycle in a rotating left-handed frame having the z_2 axis aligned with the vehicle orientation (see Figure 3.1). Equation (3.9) is another example of static input-output linearisation, with z_1 and z_2 as linearising outputs. We note also that the transformation in chained form is not unique (see, e.g. [9]).

4 Target Vehicle: SuperMARIO

The experimental comparison of the control methods to be reviewed in this chapter has been performed on the mobile robot SuperMARIO, built in the Robotics Laboratory of our Department (Figure 4.1).

4.1 Physical Description

SuperMARIO is a two-wheel differentially-driven vehicle, a mobility configuration found in many wheeled mobile robots. The two wheels have radius $r = 9.93$ cm and are mounted on the same axle of length $d = 29$ cm. The wheel radius includes also the o-ring used to prevent slippage; the rubber is stiff enough that point contact with the ground can be assumed. A small passive off-centered wheel is used as a caster, mounted in the front of the vehicle at a distance of 29 cm from the rear axle. The aluminum chassis of the robot measures $46 \times 32 \times 30.5$ cm (l/w/h) and contains two motors, transmission elements, electronics, and four 12 V batteries. The total weight of the robot (including batteries) is about 20 kg and its center of mass is located slightly in front of the rear axle. This design choice limits the disturbance on robot motion induced by sudden reorientation of the caster. Each wheel is independently driven by a DC servomotor (by MCA) supplied at 24 V with a peak torque of 0.56 Nm. Each motor is equipped with an incremental encoder counting $n_e = 200$ pulses/turn and a gearbox with reduction ratio $n_r = 20$. On-board electronics multiplies by a factor $m = 4$ the number of pulses/turn of the encoders, representing the angular increments with 16 bits.

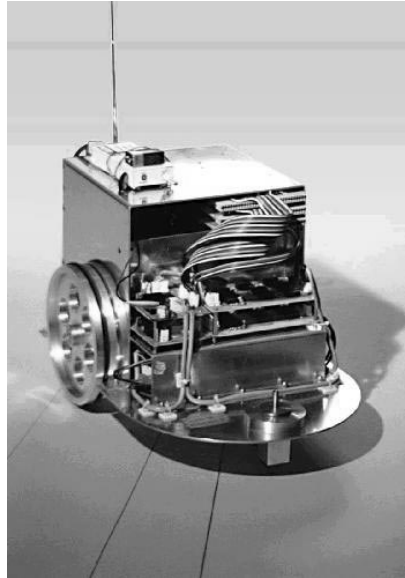


Figure 4.1. The wheeled mobile robot SuperMARIO

SuperMARIO is a low-cost prototype and presents therefore the typical nonidealities of electromechanical systems, namely friction, gear backlash, wheel slippage, actuator deadzone and saturation. These limitations clearly affect control performance. In addition, all controllers have been designed on the basis of a purely kinematic model. However, due to robot and actuator dynamics (masses and rotational inertias), velocity commands with discontinuous profile will not be exactly realised by the vehicle.

4.2 Control System Architecture

SuperMARIO has a two-level control architecture (see Figure 4.2). High-level control algorithms (including reference motion generation) are written in C++ and run with a sampling time of $T_s = 50$ ms on a remote server (a 300 MHz Pentium II), which also provides a user interface with real-time visualization as well as a simulation environment. The PC communicates through a radio modem with serial communication boards on the robot. The maximum speed of the radio link is 4800 bit/s. Wheel angular velocity commands ω_L and ω_R are sent to the robot and encoder measures $\Delta\phi_L$ and $\Delta\phi_R$ are received for odometric computations.

The low-level control layer is in charge of the execution of the high-level velocity commands. For each wheel, an 8-bit ST6265 microcontroller implements a digital PID with a cycle time of $T_c = 5$ ms. Two power amplifiers drive the motors with a 51 KHz PWM voltage.

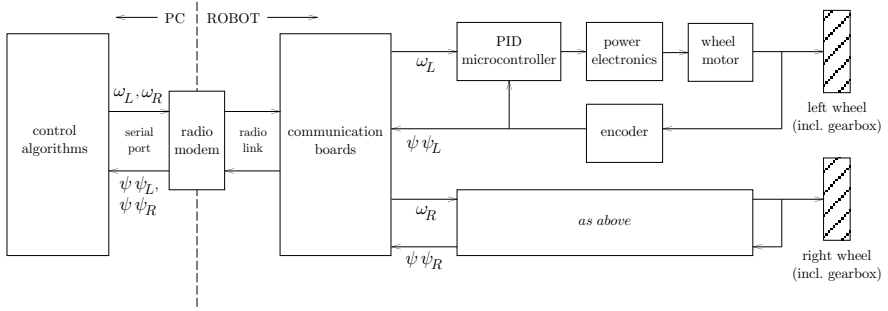


Figure 4.2. Control architecture of SuperMARIO

Custom interpolation algorithms have been developed on the PC so as to reduce the effect of quantization errors and communication delays in the reconstruction of the robot posture from the odometric information provided by the encoders. For all control schemes, an additional filtering of high-level velocity commands is included to account for vehicle and actuator dynamics. Simple first-order linear filters smooth possible discontinuities in the velocity profiles.

4.3 Kinematics

The kinematic model of SuperMARIO is given by (3.3), i.e. is equivalent to that of a unicycle. However, the actual commands are the angular velocities ω_R and ω_L of the right and left wheel, respectively, rather than the driving and steering velocities v and ω . There is, however, a one-to-one mapping between these velocities:

$$v = \frac{r(\omega_R + \omega_L)}{2} \quad \omega = \frac{r(\omega_R - \omega_L)}{d}. \quad (4.1)$$

A calibration procedure has also been developed to estimate the actual wheel radii and axle length.

The reconstruction of the current robot configuration is based on incremental encoder data (odometry). Let $\Delta\phi_R$ and $\Delta\phi_L$ be the angular wheel displacements measured during the sampling time T_s by the encoders. From (4.1), we obtain the linear and angular displacements of the robot as

$$\Delta s = \frac{r}{2} (\Delta\phi_R + \Delta\phi_L) \quad \Delta\theta = \frac{r}{d} (\Delta\phi_R - \Delta\phi_L).$$

The estimate of the posture at time $t_k = kT_s$ is computed as

$$\hat{q}_k = \begin{bmatrix} \hat{x}_k \\ \hat{y}_k \\ \hat{\theta}_k \end{bmatrix} = \hat{q}_{k-1} + \begin{bmatrix} \cos \bar{\theta}_k & 0 \\ \sin \bar{\theta}_k & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta s \\ \Delta\theta \end{bmatrix},$$

where¹

$$\bar{\theta}_k = \hat{\theta}_{k-1} + \frac{\Delta\theta}{2}.$$

Robot localization using the above odometric prediction (commonly referred to as *dead reckoning*) is accurate enough in the absence of wheel slippage and backlash. These effects are however largely reduced when the velocity is kept reasonably small and the number of backup maneuvers is limited.

4.4 Control Constraints

Because of the bounded velocity capability of the motors, each wheel can achieve a maximum angular velocity Ω . Through (4.1), we obtain bounds on the driving and steering velocities:

$$|v| \leq \Omega r, \quad |\omega| \leq \frac{2\Omega r}{d}.$$

There is, however, a more stringent constraint due to the limited resolution of the digital low-level control layer. In fact, the linear displacement resolution Δs_{min} of the robot can be computed from the previous data as

$$\Delta s_{min} = \frac{2\pi r}{m n_e n_r} = \frac{19.86 \pi}{4 \cdot 200 \cdot 20} \simeq 0.0039 \text{ cm}.$$

This value corresponds to the least significant bit of the encoder, so that the average quantization error will be less than 2 hundreds of mm. In view of the 8-bit resolution of the on-board velocity microcontroller and of the PWM circuit (having $f_m = 1/T_c = 200$ Hz as minimum pulse frequency), the actual linear velocity command has the following threshold and saturation levels

$$v_m = f_m \Delta s_{min} \simeq 0.78 \text{ cm/s} \quad v_M = v_m \cdot (2^8 - 1) = 198.9 \text{ cm/s}.$$

To prevent as much as possible wheel slippage, in our control software we have imposed the following even more conservative bounds on high-level velocity commands:

$$|v| \leq v_{max} = 0.3 \text{ m/s} \quad |\omega| \leq \omega_{max} = 0.5 \text{ rad/s}.$$

With these saturations, it is necessary to perform a suitable velocity scaling so as to preserve the curvature radius corresponding to the nominal velocities v and ω . The actual commands v_c and ω_c are then computed by defining

$$\sigma = \max \left\{ \frac{|v|}{v_{max}}, \frac{|\omega|}{\omega_{max}}, 1 \right\},$$

and letting $v_c = v$ and $\omega_c = \omega$ if $\sigma = 1$, while

$$\begin{aligned} \text{if } \sigma &= |v|/v_{max} \quad \text{then } v_c = \text{sign}(v) v_{max}, \quad \omega_c = \omega/\sigma, \\ \text{else } & v_c = v/\sigma, \quad \omega_c = \text{sign}(\omega) \omega_{max}. \end{aligned}$$

¹ Use of the average value $\bar{\theta}_k$ of the robot orientation is equivalent to the numerical integration of (3.3) via a 2nd order Runge-Kutta method.

5 Trajectory Tracking

The solution of the asymptotic tracking problem requires the combination of a nominal feedforward command with a feedback action on the error. In the control schemes to be presented, this error will be defined with respect to either the reference output trajectory (output error) or an associated reference state trajectory (state error).

5.1 Feedforward Command Generation

Assume that the representative point (x, y) of the unicycle must follow the Cartesian trajectory $(x_d(t), y_d(t))$, with $t \in [0, T]$ (possibly, $T \rightarrow \infty$). From the kinematic model (3.3) one has

$$\theta = \text{ATAN2}(\dot{y}, \dot{x}) + k\pi \quad k = 0, 1, \quad (5.1)$$

where ATAN2 is the four-quadrant inverse tangent function (undefined only if both arguments are zero). Therefore, the nominal feedforward commands are

$$v_d(t) = \pm \sqrt{\dot{x}_d^2(t) + \dot{y}_d^2(t)} \quad (5.2)$$

$$\omega_d(t) = \frac{\ddot{y}_d(t)\dot{x}_d(t) - \ddot{x}_d(t)\dot{y}_d(t)}{\dot{x}_d^2(t) + \dot{y}_d^2(t)}, \quad (5.3)$$

having differentiated (5.1) w.r.t. time in order to compute ω_d . The chosen sign for $v_d(t)$ will determine forward or backward motion of the vehicle. We note that, in order to be exactly reproducible using $v_d(t)$ and $\omega_d(t)$, the desired Cartesian motion $(x_d(t), y_d(t))$ should be twice differentiable in $[0, T]$.

A remarkable property of the unicycle is that, given an initial posture and a *consistent* desired output trajectory $(x_d(t), y_d(t))$ together with its derivatives, there is a unique associated state trajectory $q_d(t) = (x_d(t), y_d(t), \theta_d(t))$ which can be computed in a purely algebraic way², since

$$\theta_d(t) = \text{ATAN2}(\dot{y}_d(t), \dot{x}_d(t)) + k\pi \quad k = 0, 1, \quad (5.4)$$

where the value of k is chosen so that $\theta_d(0) = \theta(0)$. If $k = 1$, a backward motion will result. Therefore, the nominal orientation $\theta_d(t)$ may be computed off-line and used for defining a state trajectory error.

Note the following facts.

- When the desired linear velocity $v_d(t)$ is zero for some \bar{t} , neither the nominal angular velocity nor the nominal orientation are defined from (5.3) and (5.4), respectively. This may occur at the initial instant, if a smooth start is specified, or at a cusp along the geometric path underlying the

² This is related to the fact that (x, y) is a *flat* output for the unicycle [15] or, equivalently, a linearising output under dynamic feedback (see Section 5.4).

Cartesian trajectory $(x_d(t), y_d(t))$. For the first case, one can use (if available) higher-order differential information about $(x_d(t), y_d(t))$ at $t = 0$ in order to determine the consistent initial orientation and the initial angular velocity command. For the second case, a continuous motion is guaranteed by keeping the same orientation attained at \bar{t}^- ; by using de L'Hôpital analysis in (5.3), one can also compute the value of $\omega_d(\bar{t})$.

- More in general, the reference trajectory may be specified by separating the geometric aspects of the path (parameterised by a scalar s) from the timing law $s = s(t)$ used for path execution. The driftless nature of the kinematic model of a WMR allows to overcome in this way the above ‘zero velocity’ problem. For the unicycle, we can rewrite purely geometric relationships as

$$\begin{bmatrix} dx/ds \\ dy/ds \\ d\theta/ds \end{bmatrix} = \begin{bmatrix} \cos \theta \\ \sin \theta \\ 0 \end{bmatrix} v' + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \omega',$$

where the time commands are recovered as

$$v(t) = v'(s)\dot{s}(t) \quad \omega(t) = \omega'(s)\dot{s}(t).$$

Zero-velocity points with well-defined geometric tangent (e.g. cusps) are then obtained for $\dot{s}(\bar{t}) = 0$. The feedforward pseudo-velocity commands $v'_d(s)$ and $\omega'_d(s)$ are computed by replacing in (5.2) and (5.3) time derivatives with space derivatives.

5.2 Linear Control Design

The simplest trajectory tracking control design is based on tangent linearisation along the reference trajectory. Following [9], it is worth to reconsider the linearisation procedure of the unicycle around the trajectory. Define the state tracking error e as

$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_d - x \\ y_d - y \\ \theta_d - \theta \end{bmatrix}. \quad (5.5)$$

The difference between e and \tilde{q}_R of Section 2 is in the rotation matrix, which is computed here at the current orientation, and in a change of sign in the right hand side. Using the following nonlinear transformation of velocity inputs

$$\begin{aligned} v &= v_d \cos e_3 - u_1 \\ \omega &= \omega_d - u_2, \end{aligned} \quad (5.6)$$

the error dynamics becomes

$$\dot{e} = \begin{bmatrix} 0 & \omega_d & 0 \\ -\omega_d & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} e + \begin{bmatrix} 0 \\ \sin e_3 \\ 0 \end{bmatrix} v_d + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}. \quad (5.7)$$

Linearising (5.7) around the reference trajectory, we obtain the same linear time-varying equations (3.6), now with state e and input (u_1, u_2) .

Define the linear feedback law

$$\begin{aligned} u_1 &= -k_1 e_1 \\ u_2 &= -k_2 \operatorname{sign}(v_d(t)) e_2 - k_3 e_3. \end{aligned} \quad (5.8)$$

A desired closed-loop polynomial

$$(\lambda + 2\zeta a)(\lambda^2 + 2\zeta a\lambda + a^2) \quad \zeta, a > 0,$$

namely having *constant* eigenvalues (one negative real at $-2\zeta a$ and a complex pair with natural angular frequency $a > 0$ and damping coefficient $\zeta \in (0, 1)$) can be obtained by choosing the gains in (5.8) as

$$k_1 = k_3 = 2\zeta a \quad k_2 = \frac{a^2 - \omega_d(t)^2}{|v_d(t)|}.$$

However, k_2 will go to infinity (i.e. an infinite control effort would be required for the same transient performance) as $v_d \rightarrow 0$. Therefore, a convenient *gain scheduling* is achieved by letting $a = a(t) = \sqrt{\omega_d^2(t) + b v_d^2(t)}$ so that

$$k_1 = k_3 = 2\zeta \sqrt{\omega_d^2(t) + b v_d^2(t)} \quad k_2 = b |v_d(t)|, \quad (5.9)$$

where the factor $b > 0$ has been introduced as an additional degree of freedom. According to the controllability analysis in Section 2, these gains gracefully go to zero when local controllability around the (state) trajectory is lost because the latter stops.

In terms of the original control inputs, this design leads to the *nonlinear time-varying* controller

$$\begin{aligned} v &= v_d \cos(\theta_d - \theta) + k_1 [\cos \theta(x_d - x) + \sin \theta(y_d - y)] \\ \omega &= \omega_d + k_2 \operatorname{sign}(v_d) [\cos \theta(y_d - y) - \sin \theta(x_d - x)] + k_3(\theta_d - \theta). \end{aligned} \quad (5.10)$$

It should be emphasised that, even if the closed-loop eigenvalues are constant and with negative real part, this control law does not guarantee the asymptotic stability of the state tracking error e , because the system is still time-varying. A complete Lyapunov-based stability analysis can be however carried out by including a simple nonlinear modification, as shown hereafter.

5.3 Nonlinear Control Design

Following [26], we present now a nonlinear design for trajectory tracking. Consider again (5.7). Define

$$\begin{aligned} u_1 &= -k_1(v_d(t), \omega_d(t)) e_1 \\ u_2 &= -\bar{k}_2 v_d(t) \frac{\sin e_3}{e_3} e_2 - k_3(v_d(t), \omega_d(t)) e_3, \end{aligned} \quad (5.11)$$

with constant $\bar{k}_2 > 0$ and positive, continuous gain functions $k_1(\cdot, \cdot)$ and $k_3(\cdot, \cdot)$.

Theorem 5.1. *Assuming that v_d and ω_d are bounded with bounded derivatives, and that $v_d(t) \not\rightarrow 0$ or $\omega_d(t) \not\rightarrow 0$ when $t \rightarrow \infty$, the control law (5.11) globally asymptotically stabilises the origin $e = 0$.*

Proof. (Sketch) It is based on the use of the Lyapunov function

$$V = \frac{\bar{k}_2}{2} (e_1^2 + e_2^2) + \frac{e_3^2}{2},$$

whose time derivative along the solutions of the closed-loop system is nonincreasing since

$$\dot{V} = -k_1 \bar{k}_2 e_1^2 - k_3 e_3^2 \leq 0.$$

Thus, $\|e(t)\|$ is bounded, $\dot{V}(t)$ is uniformly continuous, and $V(t)$ tends to some limit value. Using Barbalat lemma, $\dot{V}(t)$ tends to zero. From this and analyzing the system equations, one can show that $(v_d^2 + \omega_d^2)e_i^2$ ($i = 1, 2, 3$) tends to zero so that, from the persistency of the (state) trajectory, the result follows. ■

Merging (5.5), (5.6), and (5.11), the resulting control law is

$$\begin{aligned} v &= v_d \cos(\theta_d - \theta) + k_1(v_d, \omega_d) [\cos \theta(x_d - x) + \sin \theta(y_d - y)] \\ \omega &= \omega_d + \bar{k}_2 v_d \frac{\sin(\theta_d - \theta)}{\theta_d - \theta} [\cos \theta(y_d - y) - \sin \theta(x_d - x)] + k_3(v_d, \omega_d)(\theta_d - \theta). \end{aligned} \quad (5.12)$$

Taking advantage of the previous linear analysis, we can choose the gain functions k_1 and k_3 and the constant gain \bar{k}_2 as

$$k_1(v_d(t), \omega_d(t)) = k_3(v_d(t), \omega_d(t)) = 2\zeta \sqrt{\omega_d^2(t) + b v_d^2(t)} \quad \bar{k}_2 = b,$$

with $b > 0$ and $\zeta \in (0, 1)$.

5.4 Dynamic Feedback Linearisation

A nonlinear controller for trajectory tracking based on exact dynamic feedback linearisation is now designed following [11], [12].

With reference to the general class of nonholonomic driftless systems (3.2), the dynamic feedback linearisation problem consists in finding, if possible, a dynamic state feedback compensator of the form

$$\begin{aligned} \dot{\xi} &= a(q, \xi) + b(q, \xi)u \\ w &= c(q, \xi) + d(q, \xi)u, \end{aligned} \quad (5.13)$$

with ν -dimensional state ξ and m -dimensional external input u , such that the closed-loop system (3.2)–(5.13) is equivalent, under a state transformation $z = T(q, \xi)$, to a linear controllable system.

Only necessary or sufficient (but no necessary and sufficient) conditions exist for the solution of the dynamic feedback linearisation problem. Constructive algorithms, which are essentially based on input-output decoupling, can be found in [16].

The starting point is the definition of an appropriate m -dimensional system output $\eta = h(q)$, to which a desired behaviour can be assigned (in our case, track a desired trajectory). One then proceeds by successively differentiating the output until the input appears in a nonsingular way. At some stage, the addition of integrators on a subset of the input channels may be necessary in order to avoid subsequent differentiation of the original inputs. This *dynamic extension* algorithm builds up the state ξ of the dynamic compensator (5.13). The algorithm terminates after a finite number of differentiations whenever the system is invertible from the chosen output. If the sum of the output differentiation orders equals the dimension $n + \nu$ of the extended state space, full input-state-output linearisation is also obtained. The closed-loop system is then equivalent to a set of decoupled input-output chains of integrators from u_i to η_i ($i = 1, \dots, m$).

We illustrate this exact linearisation procedure for the unicycle model (3.3). Define the linearising output vector as $\eta = (x, y)$. Differentiation w.r.t. time then yields

$$\dot{\eta} = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix},$$

showing that only v affects $\dot{\eta}$, while the angular velocity ω cannot be recovered from this first-order differential information. In order to proceed, we need therefore to add an integrator (whose state is denoted by ξ) on the linear velocity input

$$v = \xi \quad \dot{\xi} = a \quad \Longrightarrow \quad \dot{\eta} = \xi \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix},$$

being the new input a the *linear acceleration* of the unicycle. Differentiating further

$$\ddot{\eta} = \dot{\xi} \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} + \xi \dot{\theta} \begin{bmatrix} -\sin \theta \\ \cos \theta \end{bmatrix} = \begin{bmatrix} \cos \theta & -\xi \sin \theta \\ \sin \theta & \xi \cos \theta \end{bmatrix} \begin{bmatrix} a \\ \omega \end{bmatrix}$$

and the matrix multiplying the modified input (a, ω) is nonsingular provided that $\xi \neq 0$. Under this assumption, we can define

$$\begin{bmatrix} a \\ \omega \end{bmatrix} = \begin{bmatrix} \cos \theta & -\xi \sin \theta \\ \sin \theta & \xi \cos \theta \end{bmatrix}^{-1} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

so as to obtain

$$\ddot{\eta} = \begin{bmatrix} \ddot{\eta}_1 \\ \ddot{\eta}_2 \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = u. \quad (5.14)$$

The resulting dynamic compensator is

$$\begin{aligned}\dot{\xi} &= u_1 \cos \theta + u_2 \sin \theta \\ v &= \xi \\ \omega &= \frac{u_2 \cos \theta - u_1 \sin \theta}{\xi}.\end{aligned}\tag{5.15}$$

Since the dynamic compensator is one-dimensional, we have $n+\nu = 3+1 = 4$, equal to the total number of output differentiations in (5.14). Therefore, in the new coordinates

$$\begin{aligned}z_1 &= x \\ z_2 &= y \\ z_3 &= \dot{x} = \xi \cos \theta \\ z_4 &= \dot{y} = \xi \sin \theta,\end{aligned}\tag{5.16}$$

the extended system is fully linearised in a controllable form and described by the two chains of second-order input-output integrators given by (5.14), rewritten as

$$\begin{aligned}\ddot{z}_1 &= u_1 \\ \ddot{z}_2 &= u_2.\end{aligned}\tag{5.17}$$

Note that the dynamic feedback linearising controller (5.15) has a potential singularity at $\xi = v = 0$, i.e. when the unicycle is not rolling. The occurrence of such singularity in the dynamic extension process has been shown to be structural for nonholonomic systems [12]. This difficulty must be obviously taken into account when designing control laws on the equivalent linear model.

Assume the robot must follow a smooth output trajectory $(x_d(t), y_d(t))$ which is persistent, i.e. such that the nominal control input $v_d = (\dot{x}_d^2 + \dot{y}_d^2)^{1/2}$ along the trajectory does never go to zero. On the equivalent linear and decoupled system (5.17), it is straightforward to design a globally exponentially stabilizing feedback for the desired trajectory (with linear Cartesian transients) as

$$\begin{aligned}u_1 &= \ddot{x}_d(t) + k_{p1}(x_d(t) - x) + k_{d1}(\dot{x}_d(t) - \dot{x}) \\ u_2 &= \ddot{y}_d(t) + k_{p2}(y_d(t) - y) + k_{d2}(\dot{y}_d(t) - \dot{y}),\end{aligned}\tag{5.18}$$

with PD gains chosen as $k_{pi} > 0$, $k_{di} > 0$, for $i = 1, 2$.

In the implementation, velocities \dot{x} and \dot{y} can be computed via the last two expressions in (5.16), as a function of the robot state and of the compensator state ξ . Alternatively, one can use estimates of \dot{x} and \dot{y} obtained from odometric measurements. This solution is more robust with respect to unmodelled dynamics.

We conclude the discussion on trajectory tracking via dynamic feedback linearisation by offering some remarks:

- The state of the dynamic compensator should be correctly initialized at the value $\xi(0) = v_d(0)$. This guarantees exact trajectory tracking for a matched initial state of the robot. In this case, the control law (5.15–5.18) reduces to the pure feedforward action.
- Being based purely on an output tracking error definition, this method requires neither the explicit computation of $\theta_d(t)$ nor the measure of the orientation angle $\theta(t)$.
- Even for smooth persistent trajectories, problems may arise if the *actual* command $v = \xi$ crosses zero during an initial transient. However, this situation can be avoided by suitably choosing the initial state of the dynamic compensator. For example, a simple way to keep the actual commands bounded is to reset the state ξ whenever its value falls below a given threshold. This strategy results in an input command v with isolated discontinuities with respect to time.

5.5 Experiments

In the following we report experimental results of SuperMARIO following the eight-shaped reference trajectory of Figure 5.1, defined by

$$x_d(t) = \sin \frac{t}{10} \quad y_d(t) = \sin \frac{t}{20} \quad t \in [0, T].$$

The trajectory starts from the origin with $\theta_d(0) = \pi/6$ rad. A full cycle is completed in $T = 2\pi \cdot 20 \approx 125$ s. Note that the reference initial velocities are

$$v_d(0) \simeq 0.1118 \text{ m/s} \quad \omega_d(0) = 0 \text{ rad/s}.$$

In the first set of experiments, the robot configuration is initially matched with the desired reference trajectory (i.e. with initial state $q(0) = q_d(0)$). Therefore, the feedforward commands (5.2), (5.3) would allow exact trajectory following in ideal conditions. However, if the unicycle starts at rest and non-zero high-level commands $v_d(0)$ and $\omega_d(0)$ are sent to the robot, due to the actuation/vehicle dynamics there will be some transient before these velocities are actually achieved at the physical low level. The addition of a feedback action to the feedforward command is also effective in recovering the induced state error.

Figures 5.2–5.4 show the results obtained with the linearly designed controller (5.10), using the gains (5.9) with $\zeta = 0.7$ and $b = 10$. The tracking of the reference trajectory of Figure 5.1 is indeed quite accurate. Residual errors are mainly due to quantisation and discretisation of velocity commands, as well as to other nonidealities. In particular, there is a large transient error due to the vehicle/actuator dynamics because of the initial non-zero value of $v_d(0)$. This is clearly shown in Figure 5.5 which shows the norm $\|e_p\|$ of the Cartesian error.

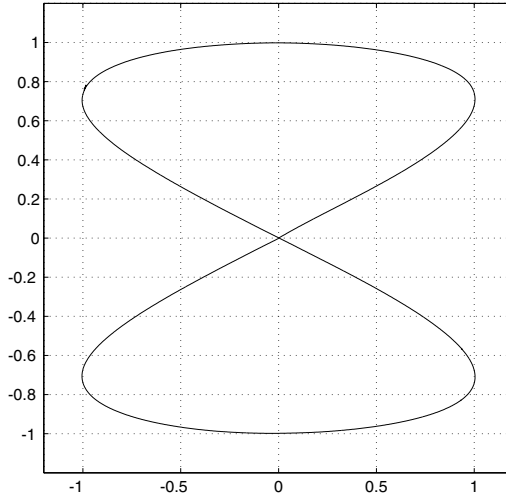


Figure 5.1. An eight-shaped reference trajectory

Similar performance is obtained with the nonlinear controller (5.12), using the same gains as above, and with the dynamic feedback linearisation controller (5.15), choosing the PD gains in (5.18) as $k_{p1} = k_{p2} = 1$, $k_{d1} = k_{d2} = 0.7$ and initializing the dynamic compensator at $\xi(0) = v_d(0)$. To appreciate the slight improvement in performance, compare the norm $\|e_p\|$ of the Cartesian error in Figures 5.6 and 5.7 with the previous result in Figure 5.5. It is found that the average error is reduced from a value of 1 cm (linear design) to 0.5 cm (nonlinear design) and finally to 0.38 cm (dynamic feedback linearisation design).

A second set of experiments was performed letting $q(0) = (0.2, -0.3, \pi/3)$ (m,m,rad), i.e. starting with an initial state error with respect to the assigned trajectory. Only the linearly designed controller and the dynamic feedback controller were compared (see Figures 5.8–5.11), using the same previous settings of control parameters. The obtained transients are quite similar, although a smaller overshoot is experienced with dynamic feedback linearisation, as implied by the choice of the PD gains.

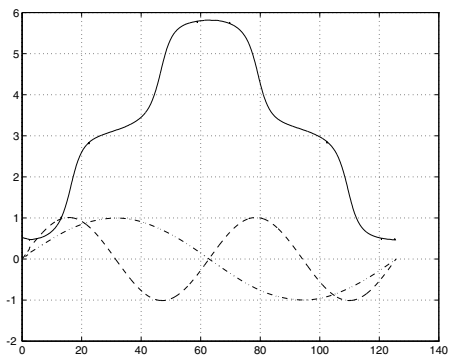


Figure 5.2. Trajectory tracking with linear feedback design: x ($--$), y ($-\cdot$) (m), and θ ($—$) (rad) vs. time (s)

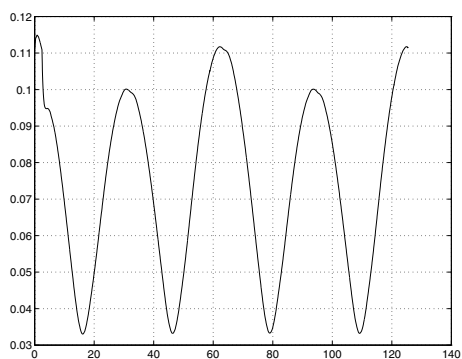


Figure 5.3. Trajectory tracking with linear feedback design: driving velocity v (m/s)

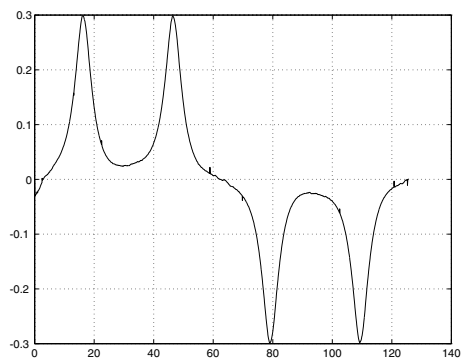


Figure 5.4. Trajectory tracking with linear feedback design: steering velocity ω (rad/s)

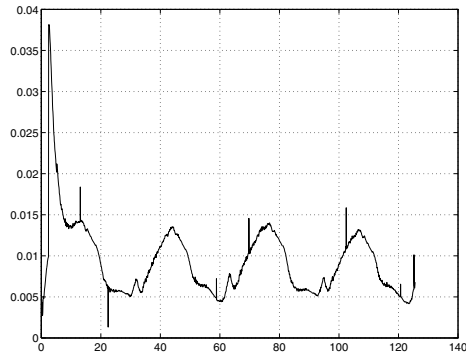


Figure 5.5. Trajectory tracking with linear feedback design: norm of Cartesian error (m)

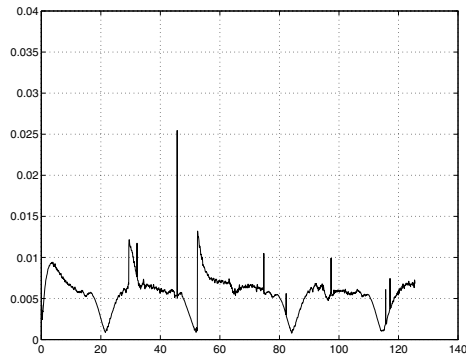


Figure 5.6. Trajectory tracking with nonlinear feedback design: norm of Cartesian error (m)

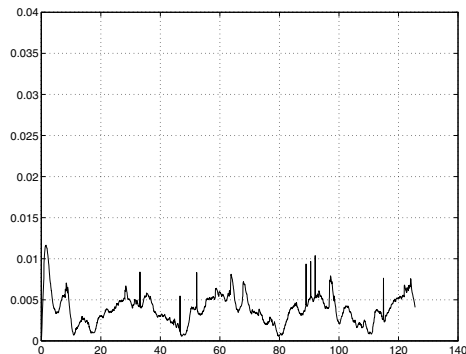


Figure 5.7. Trajectory tracking via dynamic feedback linearisation: norm of Cartesian error (m)

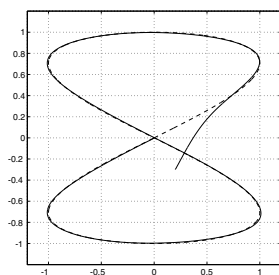


Figure 5.8. Asymptotic trajectory tracking with linear feedback design: Cartesian motion (x, y) (m)

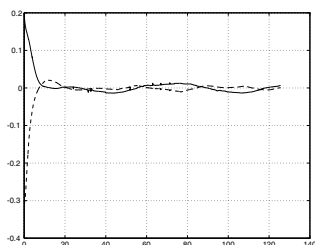


Figure 5.9. Asymptotic trajectory tracking with linear feedback design: Cartesian errors e_x and e_y (m)

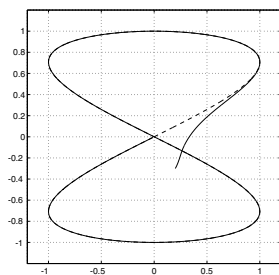


Figure 5.10. Asymptotic trajectory tracking via dynamic feedback linearisation: Cartesian motion (x, y) (m)

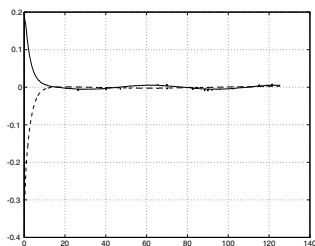


Figure 5.11. Asymptotic trajectory tracking via dynamic feedback linearisation: Cartesian errors e_x and e_y (m)

6 Posture Stabilisation

As mentioned in Section 3, posture stabilisation for nonholonomic WMRs cannot be achieved by smooth static feedback. We present two controllers based on time-varying feedback and two based on discontinuous feedback.

6.1 Smooth Time-varying Control

It has been shown in Section 5.3 that asymptotic stabilisation of a state tracking error can be achieved provided that $v_d(t)$ and $\omega_d(t)$ — which introduce a time-varying signal in the feedback control law — do not both vanish in finite time. This observation suggests that a solution to the posture stabilisation problem can be obtained by designing a desired motion, to be used as a fictitious time-varying reference, which asymptotically vanishes at the origin.

Following [26], the structure of a smooth time-varying stabilizing controller is the same of the nonlinear trajectory tracking controller (5.11):

$$\begin{aligned} u_1 &= -k_1(v_d(t), \omega_d(t)) e_1 \\ u_2 &= -\bar{k}_2 v_d(t) \frac{\sin e_3}{e_3} e_2 - k_3(v_d(t), \omega_d(t)) e_3, \end{aligned} \quad (6.1)$$

with the same notation used in Section 5.3 (in particular, (5.12) is used to generate v and ω). In this case, however, the desired trajectory is itself an additional degree of freedom, subject to the constraint that, under control (6.1), both the state tracking error e and the desired trajectory are asymptotically stabilised to the origin. A simple solution is to set, for all times t , $y_d(t) = 0$ and $\theta_d(t) = 0$ (and thus $\omega_d(t) = 0$), having only x_d in motion. A class of desired velocities is given then by

$$v_d(t) = \dot{x}_d(t) = -k_4 x_d(t) + g(e, t) \quad k_4 > 0, \quad (6.2)$$

where $g(e, t)$ is a C^2 -function uniformly bounded with respect to t , together with its partial derivative, and such that:

- A1. $g(0, t) = 0$, for all t ;
- A2. there exists a diverging sequence of instants $\{t_i\}_{i \in \mathbb{N}}$ and a continuous function $\alpha(\cdot)$ for which

$$\|e\| > \varepsilon > 0 \quad \Rightarrow \quad \left(\frac{\partial g}{\partial t}(e, t_i) \right)^2 > \alpha(\varepsilon) > 0 \quad \forall t_i.$$

Theorem 6.1. *The smooth time-varying controller (6.1), with $v_d(t)$ given by (6.2) with A1 and A2, globally asymptotically stabilises $e = 0$ and $x_d = 0$.*

Proof. (Sketch) Using the same Lyapunov function as in Theorem 5.1, the tracking error $e(t)$ is shown to be bounded. Therefore, (6.2) is a stable linear system subject to an additive bounded perturbation $g(e, t)$, so that $x_d(t)$ and $v_d(t)$ (as well as $\dot{v}_d(t)$) remain bounded. Theorem 5.1 applies and the rest of the proof uses assumptions A1 and A2 to show first that $v_d(t)$ tends to zero, then that also $\frac{\partial g}{\partial t}(e, t)$ does, and concluding the convergence of e to zero. Finally, using A2, (6.2) implies that also $x_d(t)$ tends to zero. ■

The *heating* function $g(e, t)$ plays a key role in guaranteeing asymptotic stability by sustaining motion as long as the error e is not zero, but it also determines the transient behaviour. Possible choices satisfying A1 and A2 include $g(e, t) = \|e\|^2 \sin t$ and, if the functions $k_1(\cdot, \cdot)$ and $k_3(\cdot, \cdot)$ are strictly positive,

$$g(e, t) = \frac{\exp(k_5 e_2) - 1}{\exp(k_5 e_2) + 1} \sin t \quad k_5 > 0. \quad (6.3)$$

We tested the performance of the time-varying control (6.1), with desired motion given by (6.2), initialised at $x_d(0) = 0$, and heating function (6.3). The gains have been selected as $k_1 = 0.5$, $k_2 = 2$, $k_3 = 1$, $k_4 = 1$, and $k_5 = 50$.

A forward parking task from $q(0) = (-1, -1, 0)$ (m,m,rad) to the origin is assigned. This task will be used as a baseline for comparison of all posture stabilisation methods. Figures 6.1–6.4 show that, after a relatively fast approach, convergence of the smooth time-varying controller becomes extremely slow when the unicycle is close to the goal. In particular, this is evident in Figure 6.2, a stroboscopic view of the robot motion sampled every 10 s. An inherent limitation of this control design is the large number of backup maneuvers, executed with the unicycle approximately aligned with the final desired orientation. The oscillation range of x is contained in the range $(-1, 1)$, thanks to the large value of k_5 . The steering velocity command becomes very small in the final phase.

The accuracy in regulation to the origin is determined by the satisfaction of the following terminal bounds, used in all the posture stabilisation experiments:

$$|x| < 0.2 \text{ cm} \quad |y| < 0.2 \text{ cm} \quad |\theta| < 0.02 \text{ rad}.$$

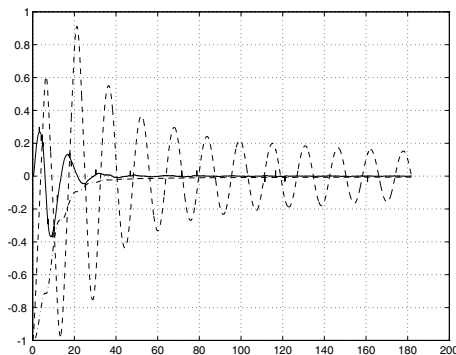


Figure 6.1. Posture stabilisation with smooth time-varying feedback: x (---), y (···) (m), and θ (—) (rad) vs. time (s)

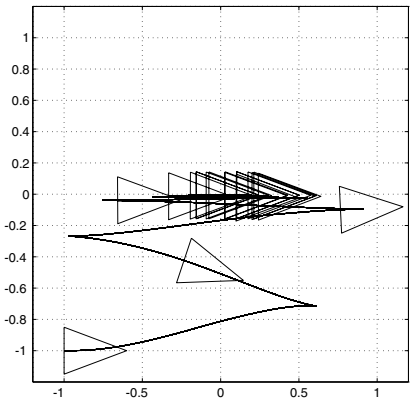


Figure 6.2. Posture stabilisation with smooth time-varying feedback: Cartesian motion (x, y) (m)

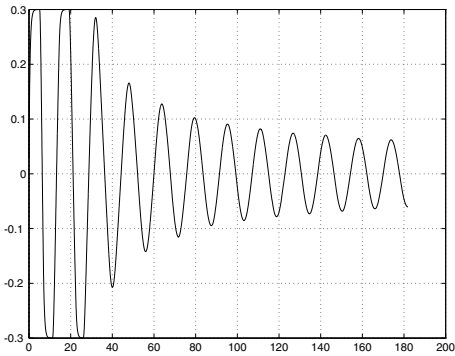


Figure 6.3. Posture stabilisation with smooth time-varying feedback: driving velocity v (m/s)

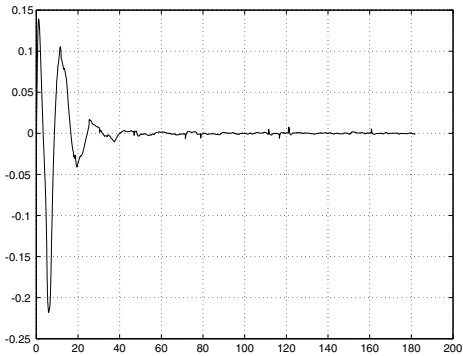


Figure 6.4. Posture stabilisation with smooth time-varying feedback: steering velocity ω (rad/s)

6.2 Nonsmooth Time-varying Control

By giving up the smoothness requirement, several controllers have been proposed for posture stabilisation with improved transient performance. We review here one of the first such designs [32], which applies to nonholonomic systems that can be transformed into chained form. The control law is continuous in time but nonsmooth with respect to the state, which is fed back only at uniformly sampled instants.

Consider the chained-form (3.9) as an equivalent unicycle model, and note that the origin of the (x, y, θ) configuration space (which is the desired posture q_d) maps into the origin of the (z_1, z_2, z_3) space. If the input u_1 is a predefined function of time, $z_{23} = [z_2 \ z_3]^T$ satisfies a linear time-varying equation driven by the input u_2 . The command u_1 is obtained by combining a simple open-loop command, which is updated as a function of the state only on a discrete-time basis, with a time-varying exogenous signal, in such a way that z_1 converges exponentially to zero when $\|z_{23}\|$ does. The other input u_2 is chosen so that $\|z_{23}\|$ converges to zero with an exponential rate.

In particular, let a sequence of uniformly spaced instants $\{t_0, t_1, t_2, \dots\}$ be defined as

$$t_h = hT \quad T = t_{h+1} - t_h > 0.$$

Define the control u_1 as

$$u_1(t) = k(z(t_h))f(t) \quad \text{for } t \in [t_h, t_{h+1}). \quad (6.4)$$

This input is a function of the state z at time $t = t_h$, while during the interval (t_h, t_{h+1}) it is defined in an open-loop fashion. Choosing a smooth and periodic function $f(t)$, such as

$$f(t) = \frac{1 - \cos \omega t}{2} \quad \omega = \frac{2\pi}{T}, \quad (6.5)$$

$u_1(t)$ is guaranteed to be continuous in time. The control function $k(z(t_h))$ is given by

$$k(z(t_h)) = -\beta [z_1(t_h) + \text{sgn}(z_1(t_h)) \gamma(\|z_{23}(t_h)\|)], \quad (6.6)$$

where

$$\text{sgn}(z_1) = \begin{cases} 1, & \text{if } z_1 \geq 0, \\ -1, & \text{if } z_1 < 0, \end{cases}$$

and

$$\beta = \frac{1}{\int_{t_h}^{t_{h+1}} f(\tau) d\tau} = \frac{2}{T} \quad \gamma(\|z_{23}\|) = \kappa \|z_{23}\|^{\frac{1}{2}} \quad \kappa > 0.$$

The control law for u_2 is designed based on the *backstepping* principle [30]. Assume that the variable z_2 in the third equation of the chained form

$$\dot{z}_3 = z_2 u_1(t) \quad (6.7)$$

is a ‘dummy’ control input z_2^d . In order to stabilise this subsystem, we choose

$$z_2^d = -\frac{\lambda_3 f^3(t) z_3}{u_1(t)} = -\frac{\lambda_3}{k(z(t_h))} f^2(t) z_3 \quad \lambda_3 > 0. \quad (6.8)$$

It can be shown that z_3 exponentially converges to zero with a rate depending on λ_3 . In the presence of a transient difference $\tilde{z}_2 = z_2 - z_2^d$, one can also prove that z_3 exponentially converges to zero if \tilde{z}_2 does. Therefore, u_2 should be designed so as to make z_2 converge to z_2^d , enforcing thus the desired behaviour for z_3 . Since the second equation of the chained form is

$$\dot{z}_2 = u_2,$$

this is accomplished, with exponential convergence, by choosing

$$u_2 = -\lambda_2(z_2 - z_2^d) + \dot{z}_2^d \quad \lambda_2 > 0,$$

where, from (6.8),

$$\dot{z}_2^d = -\lambda_3 \left(2f(t)\dot{f}(t) \frac{z_3}{k(z(t_h))} + f^3(t)z_2 \right),$$

in which we have used (6.4), (6.7), and the fact that $k(z(t_h))$ is constant over $[t_h, t_{h+1})$.

The complete nonsmooth time-varying controller is then

$$\begin{aligned} u_1 &= -k(z(t_h))f(t) \\ u_2 &= -(\lambda_2 + \lambda_3 f^3(t))z_2 - \lambda_3(\lambda_2 f^2(t) + 2f(t)\dot{f}(t)) \frac{z_3}{k(z(t_h))}. \end{aligned} \quad (6.9)$$

Equation (6.9) should be used in conjunction with (3.8) in order to generate the actual velocity inputs v and ω .

Defining a *class* \mathcal{K} function as a strictly increasing function $h : \mathbb{R}^+ \mapsto \mathbb{R}^+$ such that $h(0) = 0$, the main result is summarised in the following theorem.

Theorem 6.2. *Consider the unicycle in chained form (3.9) under the control (6.9), with the definitions (6.5) and (6.6). Then, the origin $z = 0$ is \mathcal{K} -exponentially stable, i.e. there exist a constant $\lambda_z > 0$ and a class \mathcal{K} function $h_z(\cdot, T)$ such that*

$$\|z(t)\| \leq h_z(\|z(0)\|, T) e^{-\lambda_z t} \quad \forall z(0) \in \mathbb{R}^3, \forall t \geq 0.$$

Proof. It is a specialisation of the general proof for n -dimensional chained-form systems, see [32]. ■

Figures 6.5–6.8 show the results of the application of the control law (6.9), with $T = 6$ s and $\kappa = 0.8$, $\lambda_2 = \lambda_3 = 0.4$, for executing the baseline forward parking task. The rate of convergence of the nonsmooth time-varying controller is somewhat improved but still quite slow. A stroboscopic view of the unicycle motion sampled every 5 s is reported in Figure 6.6. Note that the approach in the y direction is very uniform, while maneuvers in the vicinity of the goal are aimed at adjusting θ rather than x . This is intrinsic in the structure of the chained form used for the control design.

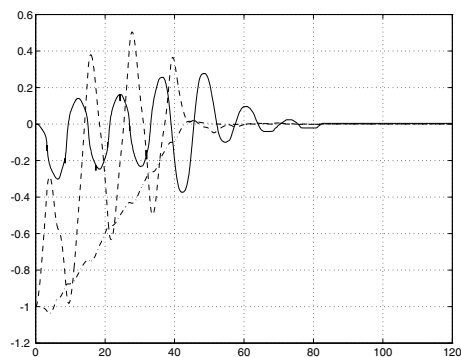


Figure 6.5. Posture stabilisation with nonsmooth time-varying feedback: x (---), y (-·-) (m), and θ (—) (rad) vs. time (s)

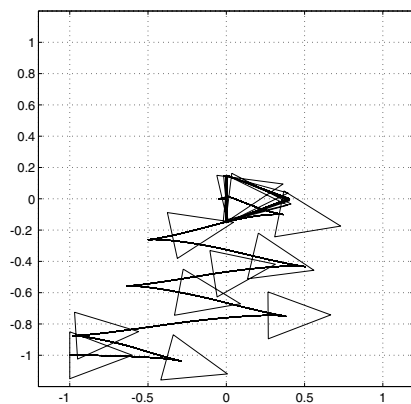


Figure 6.6. Posture stabilisation with nonsmooth time-varying feedback: Cartesian motion (x, y) (m)

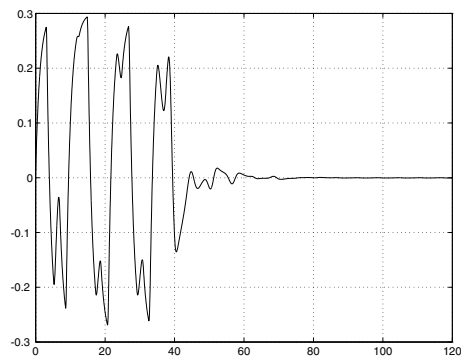


Figure 6.7. Posture stabilisation with nonsmooth time-varying feedback: driving velocity v (m/s)

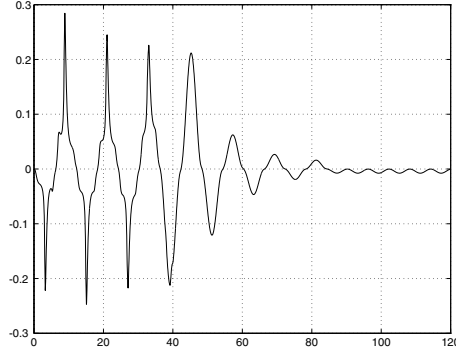


Figure 6.8. Posture stabilisation with nonsmooth time-varying feedback: steering velocity ω (rad/s)

6.3 Control Based on Polar Coordinates

Another technique which allows to overcome the obstruction of Brockett theorem is to apply a change of coordinates such that the input vector fields of the transformed equations are singular at the origin. This approach has been proposed in [1], where a Lyapunov-like design of a posture stabilizing controller is carried out using a polar coordinate transformation. The control law, once rewritten in terms of the original state variables, is discontinuous at the origin of the configuration space \mathcal{Q} .

With reference to Figure 6.9, we define the following set of *polar* coordinates for the unicycle. Let ρ be the distance of the reference point (x, y) of the unicycle from the goal (the origin), γ be the angle of the pointing vector to the goal w.r.t. the unicycle main axis, and δ be the angle of the same pointing vector w.r.t. the x axis (orientation error), i.e.

$$\begin{aligned}\rho &= \sqrt{x^2 + y^2} \\ \gamma &= \text{ATAN2}(y, x) - \theta + \pi \\ \delta &= \gamma + \theta.\end{aligned}$$

Angles γ and δ are undefined for $x = y = 0$. In practice, during the vehicle motion, when $\rho \rightarrow 0$ one must retain the values for γ and δ assumed in the final approaching phase. In these coordinates, the unicycle equations become

$$\begin{aligned}\dot{\rho} &= -\cos \gamma v \\ \dot{\gamma} &= \frac{\sin \gamma}{\rho} v - \omega \\ \dot{\delta} &= \frac{\sin \gamma}{\rho} v.\end{aligned}\tag{6.10}$$

The input vector field associated to v is singular for $\rho = 0$.

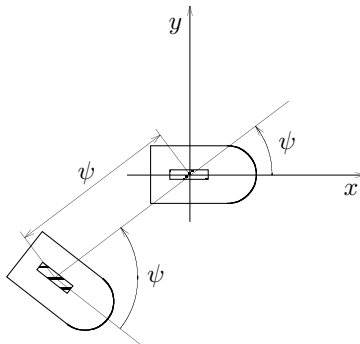


Figure 6.9. Definition of polar coordinates for the unicycle

In order to achieve the goal posture, variables e , γ , and δ should all converge to zero. This is guaranteed by the following result, adapted from [1].

Theorem 6.3. *Consider the polar coordinate description (6.10) of the unicycle and the feedback control*

$$\begin{aligned} v &= k_1 \rho \cos \gamma \\ \omega &= k_2 \gamma + k_1 \frac{\sin \gamma \cos \gamma}{\gamma} (\gamma + k_3 \delta), \end{aligned} \quad (6.11)$$

with k_1 and k_2 positive constants. The closed-loop system (6.10)–(6.11) is then globally asymptotically driven to the posture $(\rho, \gamma, \delta) = (0, 0, 0)$.

Proof. (Sketch) Consider the function

$$V = \frac{1}{2} (\rho^2 + \gamma^2 + k_3 \delta^2),$$

as a candidate Lyapunov function. Its time derivative along the solutions of the closed-loop system is nonincreasing since

$$\dot{V} = -k_1 \cos^2 \gamma \rho^2 - k_2 \gamma^2 \leq 0.$$

Thus, the state is bounded in norm, $\dot{V}(t)$ is uniformly continuous, and $V(t)$ tends to a limit value. By Barbalat lemma, $\dot{V}(t)$ tends to zero and thus also ρ and γ do. Analyzing the closed-loop system, one can show that $\dot{\rho}$ and $\dot{\delta}$ converge to zero, that δ converges to some finite limit $\bar{\delta}$ while $\dot{\gamma}$ tends to the finite limit $-k_1 k_3 \bar{\delta}$ and is uniformly continuous. Again through Barbalat lemma, this finite limit must be zero and thus also δ converges to zero. ■

Figures 6.10–6.13 refer to the results of the application of controller (6.11), with gains $k_1 = 1$, $k_2 = 3$ and $k_3 = 2$, for executing the baseline forward parking task. The convergence to the goal is very fast and natural. In Figure 6.11, a stroboscopic view of the unicycle motion sampled every 1 s is given. Both velocity commands are saturated for 2 to 3 seconds.

In the second experiment, a parallel parking task from $q(0) = (0, -1.5, 0)$ (m,m,rad) to the origin of the configuration space is assigned. The same gains as in the previous experiment have been used. The obtained results are shown in Figures 6.14–6.17 and indicate that there is no backup maneuver in this case. If the robot had been initially closer to the positive x axis, the controller (6.11) would have automatically driven the robot backwards and then in forward motion to the goal. This is a general property of the controller: in the final phase, the vehicle will always approach the goal in forward motion, having executed at most one backup maneuver.

We finally note that the behaviour of the controlled system is not continuous with respect to the initial state. For example, assume that the initial configuration is $x(0) = a > 0$, $y(0) = \varepsilon$ and $\theta(0) = 0$. Positive and negative arbitrarily small values of ε will lead to different transient motions to the goal (in fact, symmetric with respect to the x axis).

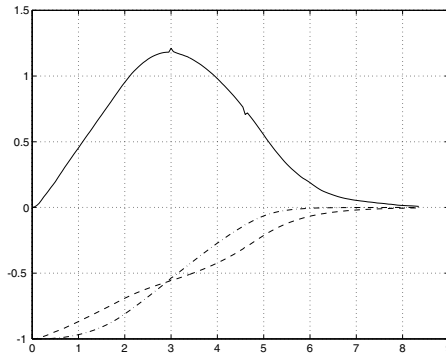


Figure 6.10. Posture stabilisation using feedback in polar coordinates (forward parking): x (—), y (—·), and θ (—) (rad) vs. time (s)

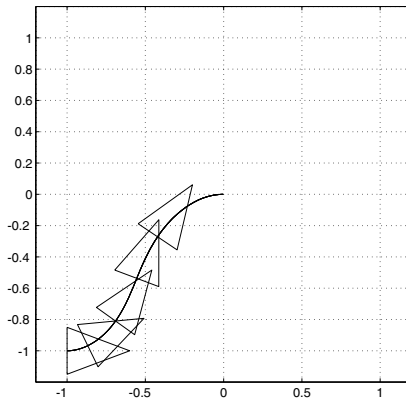


Figure 6.11. Posture stabilisation using feedback in polar coordinates (forward parking): Cartesian motion (x, y) (m)

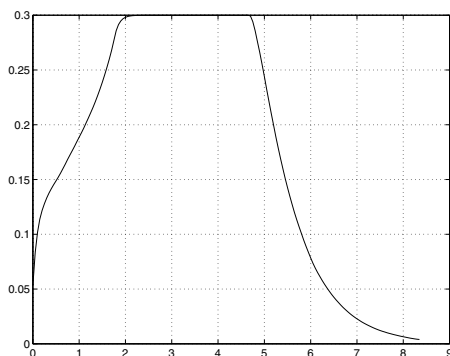


Figure 6.12. Posture stabilisation using feedback in polar coordinates (forward parking): driving velocity v (m/s)

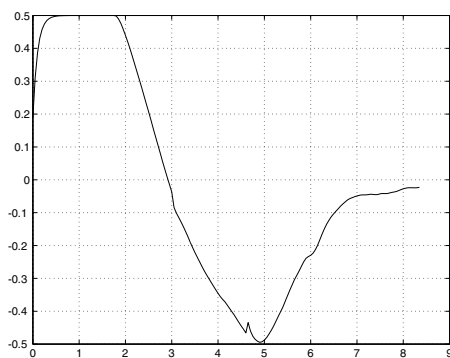


Figure 6.13. Posture stabilisation using feedback in polar coordinates (forward parking): steering velocity ω (rad/s)

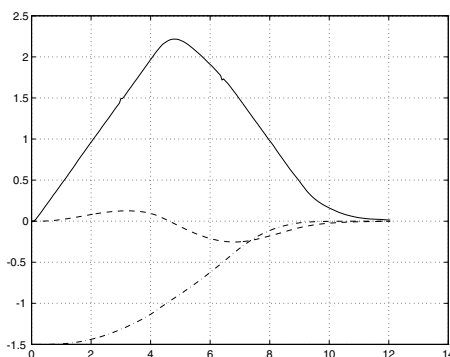


Figure 6.14. Posture stabilisation using feedback in polar coordinates (parallel parking): x (---), y (··) (m), and θ (—) (rad) vs. time (s)

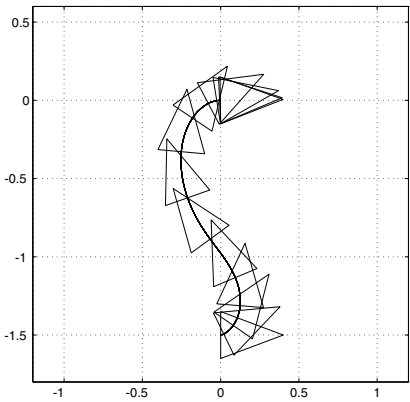


Figure 6.15. Posture stabilisation using feedback in polar coordinates (parallel parking): Cartesian motion (x, y) (m)

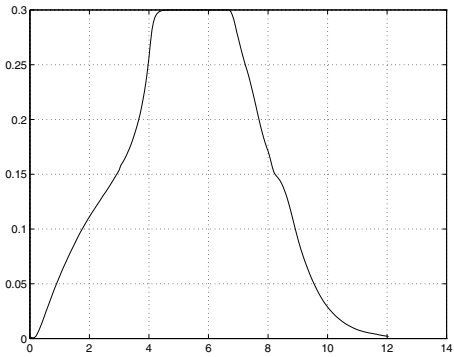


Figure 6.16. Posture stabilisation using feedback in polar coordinates (parallel parking): driving velocity v (m/s)

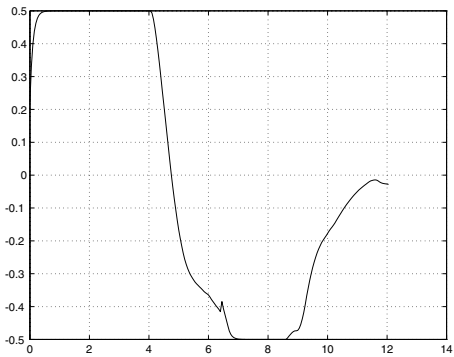


Figure 6.17. Posture stabilisation using feedback in polar coordinates (parallel parking): steering velocity ω (rad/s)

6.4 Dynamic Feedback Linearisation

Following [14], we show how to extend the trajectory tracking controller based on dynamic feedback linearisation to address the posture stabilisation problem, while avoiding the intrinsic singularity that occurs when the robot comes to a stop. This simply requires an appropriate choice of the PD gains and a suitable initialisation of the dynamic compensator state ξ .

We denote by

$$\mathcal{Q}^* = \{q \in \mathcal{Q} : (x = 0, \cos \vartheta \geq 0) \text{ OR } (y = 0, \cos \vartheta = -1)\}$$

a subset of \mathcal{Q} which will require special attention. The remaining part $\mathcal{Q}/\mathcal{Q}^*$ of the configuration space can be partitioned in two regions:

$$\begin{aligned}\mathcal{Q}^r &= \{q \in \mathcal{Q}/\mathcal{Q}^* : x \geq 0\} \\ \mathcal{Q}^l &= \{q \in \mathcal{Q}/\mathcal{Q}^* : x < 0\}.\end{aligned}$$

The main result is the following.

Theorem 6.4. *Consider the unicycle system (3.3) under the action of the dynamic compensator (5.15). A PD control law on the Cartesian error*

$$\begin{aligned}u_1 &= -k_{p1}x - k_{d1}\dot{x} \\ u_2 &= -k_{p2}y - k_{d2}\dot{y}\end{aligned}\tag{6.12}$$

yields exponential convergence from any initial configuration $q(0) \in \mathcal{Q}/\mathcal{Q}^$ to the origin, if the following assumptions hold:*

A1. The control gains $k_{pi} > 0$, $k_{di} > 0$ ($i = 1, 2$) satisfy the conditions

$$k_{d1}^2 - 4k_{p1} = k_{d2}^2 - 4k_{p2} > 0\tag{6.13}$$

$$k_{d2} - k_{d1} > 2\sqrt{k_{d2}^2 - 4k_{p2}}.\tag{6.14}$$

A2. The initial state of the dynamic compensator is chosen as

$$\begin{aligned}\xi(0) &< 0 & (\text{backward motion}) & & \text{if } q(0) \in \mathcal{Q}^r \\ \xi(0) &> 0 & (\text{forward motion}) & & \text{if } q(0) \in \mathcal{Q}^l,\end{aligned}$$

but its value is otherwise arbitrary, except for the additional condition

$$\xi(0) \neq 2 \frac{k_{p1}x(0) \sin \theta(0) - k_{p2}y(0) \cos \theta(0)}{k_{d2} - k_{d1}}.\tag{6.15}$$

Proof. Use of control (6.12) in (5.17) implies that the Cartesian coordinates x and y converge to zero exponentially, provided that the original control inputs v and ω given in (5.15) remain bounded. To show this, we must prove that (i) ξ does not go to zero in finite time, and (ii) ω tends to zero for $t \rightarrow \infty$, in spite of its denominator ξ vanishing.

(i) Since from (5.16) it is $\xi^2 = z_3^2 + z_4^2$, one has $\xi(\bar{t}) = 0$ iff $z_3(\bar{t}) = z_4(\bar{t}) = 0$, for a generic instant $\bar{t} \geq 0$. Integrating the closed-loop system (5.17) under control (6.12), we have

$$z_3(t) = a_{31}e^{\lambda_{11}t} + a_{32}e^{\lambda_{12}t} \quad (6.16)$$

$$z_4(t) = a_{41}e^{\lambda_{21}t} + a_{42}e^{\lambda_{22}t}, \quad (6.17)$$

where coefficients a_{kj} and eigenvalues λ_{ij} are easily obtained as functions of initial state and PD gains. From these expressions and condition (6.13), it is possible to show that a finite $\bar{t} > 0$ such that $\xi(\bar{t}) = 0$ exists iff

$$\frac{\xi(0) \cos \theta(0) - x(0)\lambda_{11}}{\xi(0) \cos \theta(0) - x(0)\lambda_{12}} = \alpha \frac{\xi(0) \sin \theta(0) - y(0)\lambda_{21}}{\xi(0) \sin \theta(0) - y(0)\lambda_{22}},$$

with $\alpha = \lambda_{11}\lambda_{22}/\lambda_{12}\lambda_{21}$. From this, a quadratic equation in $\xi(0)$ is derived which has the single nonzero root

$$\xi(0) = \frac{\lambda_{11}x(0) \sin \theta(0) + \lambda_{22}y(0) \cos \theta(0) - \alpha(\lambda_{12}x(0) \sin \theta(0) + \lambda_{21}y(0) \cos \theta(0))}{(1 - \alpha) \sin \theta(0) \cos \theta(0)}.$$

Once rewritten in terms of the PD gains, this expression leads to the forbidden initialisation condition (6.15).

(ii) First of all, it is straightforward to prove that assumption A1 implies that the eigenvalues are real and ordered as $\lambda_{11} < \lambda_{12} < \lambda_{21} < \lambda_{22} < 0$. From (5.15), we rewrite ω as

$$\omega = \frac{-z_4u_1 + z_3u_2}{\xi^2},$$

and using (6.16–6.17) and (6.12), the numerator of ω takes the form

$$-z_4u_1 + z_3u_2 = \gamma_1e^{(\lambda_{11}+\lambda_{21})t} + \gamma_2e^{(\lambda_{12}+\lambda_{21})t} + \gamma_3e^{(\lambda_{11}+\lambda_{22})t} + \gamma_4e^{(\lambda_{12}+\lambda_{22})t},$$

with $\gamma_i \in \mathbb{R}$. Its asymptotic rate of convergence is certainly larger than $2|\lambda_{22}|$ due to the eigenvalue ordering. As for the denominator, squaring and adding (6.16) and (6.17) gives

$$\xi^2 = \eta_1e^{2\lambda_{11}t} + \eta_2e^{(\lambda_{11}+\lambda_{12})t} + \eta_3e^{2\lambda_{12}t} + \eta_4e^{2\lambda_{21}t} + \eta_5e^{(\lambda_{21}+\lambda_{22})t} + \eta_6e^{2\lambda_{22}t},$$

with $\eta_i \in \mathbb{R}$. Since the asymptotic rate of convergence of this quantity is exactly $2|\lambda_{22}|$, we conclude that ω tends to zero as $t \rightarrow \infty$.

To finish the proof, it is necessary to show that also the orientation θ converges to zero. This is easily understood from the following facts:

- The unicycle reaches the origin with a horizontal tangent ($\theta = 0$ or π), because y approaches zero faster than x in view of the eigenvalue ordering.
- Motion inversions do not occur since $v = \xi$ never crosses zero, as shown in the first part of the proof.
- The trajectory is confined to the region (either \mathcal{Q}_r or \mathcal{Q}_l) from which the unicycle starts. In fact, x and y never change sign because the eigenvalues are real and thanks to the choice of sign for $\xi(0)$ in assumption A2.

Finally, also the convergence of θ to zero is exponential. Indeed, since ω goes exponentially to zero, the same is true for its integral θ . ■

Some remarks are needed at this point.

- As the Cartesian position transients are linear, the unicycle trajectories obtained with the proposed controller are completely predictable and can be easily shaped by choosing the PD control gains. Note that the unicycle can reach the goal either with a forward or with a backward motion.
- The equality part of condition (6.13) in Theorem 6.4 is by no means necessary; it is only used for deriving a closed form for the forbidden initialisation (6.15) of the dynamic compensator.
- In view of the discontinuity at the origin of the linearising controller with respect to the state (x, y, θ, ξ) of the extended system, as well as of the fact that the initial configuration should belong to $\mathcal{Q}/\mathcal{Q}^*$, the proposed feedback controller does not yield Lyapunov stability in a strict sense, but simply exponential convergence.

If the initial configuration $q(0)$ belongs to \mathcal{Q}^* , Theorem 6.4 cannot be applied. In fact, the PD control (6.12) would bring the unicycle to the origin with the wrong orientation, namely, $\theta = \pi$ if $\cos \theta(0) > 0$, $\theta = \pm\pi/2$ if $\cos \theta(0) = 0$, $\theta = -\pi$ if $\cos \theta(0) = -1$. In such a situation, it is necessary to reset the compensator state at some time $t_v > 0$, so as to invert the motion at a configuration $q \in \mathcal{Q}/\mathcal{Q}^*$. A simple way to obtain this is to introduce a via point q_v in the regulation procedure, as illustrated later in this section by the parallel parking experiment. This should not be seen as a drawback of the method; a suitable choice of the via point allows better control of the path shape while approaching the goal configuration. In particular, the resulting stabilisation motion contains at most one backup maneuver.

Figures 6.18–6.21 refer to the baseline forward parking task, using dynamic feedback linearisation plus PD control with gains $k_{p1} = 2$, $k_{d1} = 3$, $k_{p2} = 12$, $k_{d2} = 7$, and compensator initialisation $\xi(0) = v_{max} = 0.3$ (m/s). The convergence to the goal is fast and very natural, as shown in Figure 6.19, a stroboscopic view of the robot motion sampled every 1.5 s. Note that saturation occurs on both inputs during the transient phase.

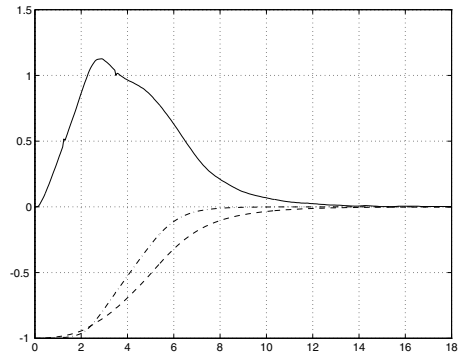


Figure 6.18. Posture stabilisation using dynamic feedback linearisation (forward parking): x ($--$), y ($- \cdot -$) (m), and θ ($—$) (rad) vs. time (s)

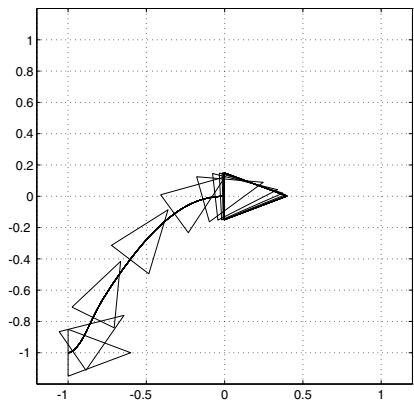


Figure 6.19. Posture stabilisation via dynamic feedback linearisation (forward parking): Cartesian motion (x, y) (m)

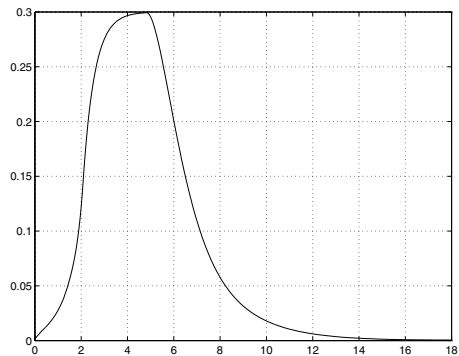


Figure 6.20. Posture stabilisation via dynamic feedback linearisation (forward parking): driving velocity v (m/s)

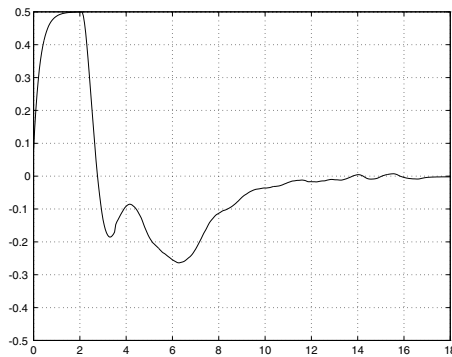


Figure 6.21. Posture stabilisation via dynamic feedback linearisation (forward parking): steering velocity ω (rad/s)

We also report the results for a parallel parking task with initial configuration given by $q(0) = (0, -1, 0)$ (m,m,rad). The via point is chosen as $q_v = (-1, -0.6, 0)$ (m,m,rad). The PD gains are the same as before, while the compensator state initialisation is chosen here as $\xi(0) = -v_{max}$ for the first phase, performed in backward motion, and $\xi(t_v) = v_{max}$ for the second phase, which is started in a neighbourhood of q_v and performed in forward motion. The results are shown in Figures 6.22–6.25.

The simple first-order linear filter introduced to account for actuator dynamics is also effective in smoothing the discontinuity in the driving velocity generated by the reset procedure. On the other hand, the presence of the same filter for the steering velocity, coupled with the software velocity saturation, neutralises the effect of the singularity in ω due to the zero crossing of the filtered driving velocity.

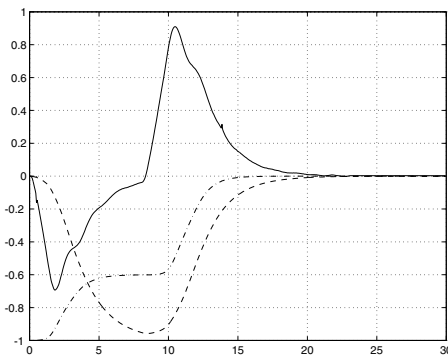


Figure 6.22. Posture stabilisation via dynamic feedback linearisation (parallel parking): x (---), y (-·-) (m), and θ (—) (rad) vs. time (s)

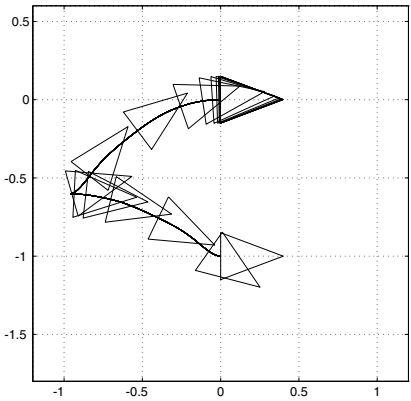


Figure 6.23. Posture stabilisation via dynamic feedback linearisation (parallel parking): Cartesian motion (x, y) (m)

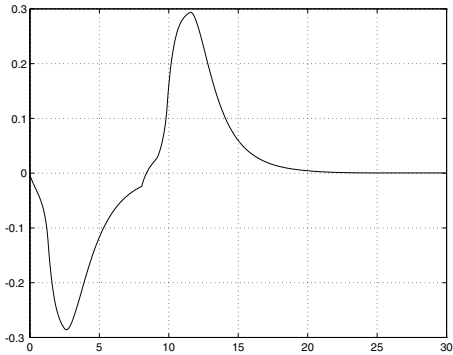


Figure 6.24. Posture stabilisation via dynamic feedback linearisation (parallel parking): driving velocity v (m/s)

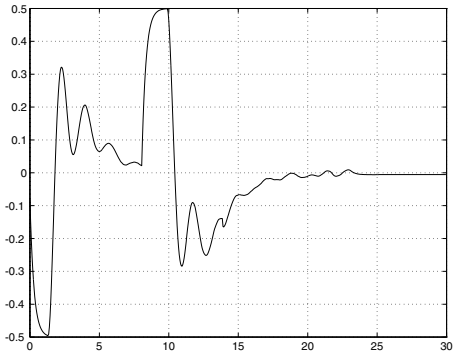


Figure 6.25. Posture stabilisation via dynamic feedback linearisation (parallel parking): steering velocity ω (rad/s)

7 Guidelines for End-users

7.1 Summary and Comparison

We have performed several motion tasks with SuperMARIO using the presented control laws. The experimental tests presented in this chapter are representative of the average performance of the controllers. We summarise hereafter our acquired experience in general observations that can be useful guidelines for implementation of the same control strategies for other vehicles.

First of all, the computational load for all methods is quite similar in the case of the unicycle. Basically, both trajectory tracking and posture stabilisation controllers can be implemented with on-board computing power. Our choice of separating high-level control routines, performed on a remote server, from the low-level control loops in charge of realizing the reference velocity commands on each wheel reflects only the choice of a modular structure. It is however expected that such decomposition would become more convenient or even mandatory for wheeled mobile robots with more complex kinematics, such as a tractor vehicle towing a number of trailers.

The three reviewed control methods for trajectory following tasks show similar performance. All of them can be generalised to more complex vehicles, provided their models are transformable in chained-form. Such generalisations can be found in [27] and [13]. From the point of view of control parameters tuning, especially for more complex WMRs, the dynamic feedback linearisation technique promises to be simpler since it always boils down to the choice of stabilizing gains for a chain of integrators [13]; in particular, it can be carried out on the original equations without resorting to the transformation in chained form.

In Table 7.1, the control results for posture stabilisation tasks are compared in terms of performance, ease of control parameters tuning, sensitivity to nonidealities, generalisability to more complex nonholonomic WMRs, and relations with the design of tracking controllers.

Time-varying controllers, both smooth and nonsmooth, exhibit a rather slow final convergence to the goal in spite of substantial progress during the first motion phase. In general, the nonsmooth controller should behave better because it achieves an exponential rate of convergence, but the dependence of this rate on the available gains is critical. The oscillatory behaviour of the vehicle during the approach to the goal, which makes the motion rather erratic, is an intrinsic characteristic of both time-varying control laws. In fact, the exogenous time dependence needed for stabilisation is introduced through the oscillatory motion of a virtual reference vehicle in the method of Section 6.1 and through the periodic time function weighting the first command in the method of Section 6.2. In any case, the presence of several motion inversions makes these methods quite sensitive to mechanical nonidealities (e.g. backlash) of the wheels. This situation may introduce a remarkable difference


	Smooth time-varying stabilisation	Nonsmooth time-varying stabilisation	Design with polar coordinates	Dynamic feedback linearisation
Achieved performance	very slow erratic	slow erratic	fast natural	fast natural
Ease of control tuning	a few parameters problematic	a few parameters critical	simple	simple PD ξ initialisation
Sensitivity to nonidealities	many backups	backups and sampled feedback	good	good integral action
Generalisation to other WMRs	yes if chained form	yes if chained form	no	yes if chained form
Relation with tracking control	extended from tracking	none	none	same PD control law

Table 7.1. A comparison of the posture stabilisation controllers presented in this chapter

between computed configuration from the internal odometry and actual displacement of the vehicle on the ground. In our experience, this behaviour was confirmed also in experiments performed with a car-like vehicle (the MARIO robot [25]), where a nonnegligible backlash on the steering angle of the front wheels led to a substantial error in the final positioning. Another potential problem with the presented nonsmooth controller is that, being based on a low-rate sampled state feedback (see (6.4)), the robot could in principle ‘miss’ the final goal even if passing through it. Among the positive features of these time-varying control laws, we mention that they can readily be generalised to more complex WMRs allowing a chained-form representation (see [27], [32]). Also, the smooth time-varying controller is a direct outgrowth of the trajectory tracking controller of Section 5.3.

The controller based on polar coordinates transformation performed very well. The resulting vehicle path is very natural (in the sense that is similar to the one followed by an experienced human driver) and practical convergence is quite fast, with a weak dependence on the choice of the few control gain parameters. Since at most one backup maneuver is needed, disturbances due to wheel backlash are minimised. Unfortunately, a direct extension of such controller is not available for vehicles with more complex kinematics. The idea of using a state-space transformation that is singular at the goal configuration, however, stands on its own and has been exploited by other researchers, e.g. in [3]. Also, the polar coordinates controller has been adapted

in [1] for stabilisation about successive via points extracted from a desired path, but in this way only approximate trajectory tracking can be obtained.

Similar positive comments can be drawn on the performance of the posture stabilisation method designed via dynamic feedback linearisation. In particular, this scheme allows also parallel parking with backward/forward motion, which is a very natural maneuver. The control tuning requires the choice from a very large feasible set of PD gains. The relationship with the analogous controller for trajectory tracking is very simple: it is sufficient to add the feedforward terms, i.e. the reference output position, velocity and acceleration (compare (5.18) with (6.12)). As for the use of an additional dynamics in the control law, it has some pros and cons. On one side, this design automatically takes into account the nonideality of a first-order kinematic model of the unicycle, by bringing linear acceleration into the picture. On the other side, it is necessary to prevent zeroing of the compensator state and the consequent singularity of the control commands; this may be achieved in practice by the simple strategy of filtering plus saturating the velocity commands. The generalisation to point-to-point motion tasks for WMRs with more complex kinematics is under way. It basically consists in extending the idea of suitably shaping the transient behaviour on the linear side of the problem by appropriate selection of the gain structure (a PD^{n-2} for n generalised coordinates), so as to achieve a smooth and correct ‘entrance’ into the goal for the two outputs representing the robot Cartesian position.

All the controllers reviewed in this chapter use a measure of the state reconstructed on the basis of the robot odometry. In principle, the actual motion of SuperMARIO on the ground may be quite different and should be computed with the aid of exteroceptive sensors. However, in our experiment, this difference could not be appreciated visually (as shown by the videos on the web page <http://labrob.ing.uniroma1.it/projects/ramsete.html>). The satisfactory performance of our dead reckoning localisation system is of course related to the execution of relatively slow motion tasks.

A final remark is needed about the application of the control methods reviewed in this chapter when workspace obstacles are present. In a completely known environment, it may be convenient to tackle the navigation problem of a WMR using a three-layer control structure. The highest layer is devoted to motion planning and takes care of the nominal avoidance of obstacles; the nonholonomic motion constraints of the WMR may or may not be taken into account at this stage. The second layer takes charge of motion execution and uses one of the trajectory tracking controllers given in this chapter. In the vicinity of the goal, fine posture regulation (docking) can be obtained at the lowest layer by means of one of the presented stabilizing controllers.

7.2 Future Directions

There are some important issues barely mentioned in this chapter that deserve further research.

Inclusion of dynamics. For massive vehicles and/or at high speeds, consideration of robot dynamics is necessary for realistic control design. The dynamics of general nonholonomic systems is thoroughly analysed in [5] and, more specifically for WMRs, in [7]. Interestingly, nonlinear static state feedback can be used to cancel, in the nominal case, all inertial parameters so as to lead to a purely (second-order) kinematic model of the form

$$\dot{q} = G(q)w, \quad \dot{w} = a,$$

in place of (3.2), with (q, w) as the $(n+m)$ -dimensional state and acceleration a as the new control input. The control laws used in this chapter do not directly apply to this case (they may have finite jumps in the velocity), but it is relatively easy to rework feasible modifications.

Robust control design. Very few papers have explicitly addressed robustness issues in the control of nonholonomic systems. Robust stabilisation of WMRs in chained form was obtained in [4] and [19] by applying iteratively an open-loop command; exponential convergence to the desired equilibrium is obtained for small perturbations in the kinematic model. Another possible approach to the design of effective control laws in the presence of nonidealities and uncertainties is represented by learning control, as shown in [25]. We also note that perturbations acting on nonholonomic mobile robots are not of equal importance: a deviation in a direction compatible with the vehicle mobility (e.g. slippage of the wheels on the ground) is clearly not as severe as a deviation which violates the kinematic constraints of the system (e.g. lateral sliding).

Use of exteroceptive feedback. Proprioceptive sensors, such as incremental encoders, are obviously unreliable in the presence of wheel slippage. As a result, the robot may progressively ‘lose’ itself in the environment. A possible solution is to close the feedback loop using exteroceptive sensors, which provide absolute information about the robot localisation in its workspace; sensor fusion techniques are relevant at this stage. The design of sensor-level controllers for nonholonomic robots is at the beginning stage but growing fast. An example of an on-board visual servoing for trajectory tracking is presented in [20].

Extension to WMRs not transformable in chained form. Among the open problems in motion control of general WMRs, we mention the case of multi-body vehicles that do not admit a transformation in chained form, such as a unicycle or car-like tractor with two or more trailers hooked at a non-zero distance from the axle of the previous moving body. A possible approach to posture stabilisation, using iterative steering of a nilpotent approximation model, can be found in [33].

Acknowledgements

We wish to thank here Prof. Giovanni Ulivi of Third University of Rome for participating in the design and development of SuperMARIO. We also gratefully acknowledge the cooperation of all the students who have contributed to this work. R. Ascione designed the mechanical and electronic parts of SuperMARIO. L. Alivernini developed a first set of control routines in Visual Basic. M. Palmieri developed the structure of the C++ control software. M. Laurelli brought the software to the current state.

References

1. Aicardi M, Casalino G, Bicchi A, Balestrino A 1995 Closed loop steering of unicycle-like vehicles via Lyapunov techniques. *IEEE Robotics and Automation Magazine* 2(1):27–35
2. Alexander J C, Maddocks J H 1989 On the kinematics of wheeled mobile robots. *International Journal of Robotics Research* 8(5):15–27
3. Astolfi A 1996 Discontinuous control of nonholonomic systems. *Systems and Control Letters* 27:37–45
4. Bennani M K, Rouchon P 1995 Robust stabilization of flat and chained systems. In: *Proceedings of 3rd European Control Conference* Rome, I, pp 2642–2646
5. Bloch A M, Reyhanoglu M, McClamroch N H 1992 Control and stabilization of nonholonomic dynamic systems. *IEEE Transactions on Automatic Control* 37:1746–1757
6. Brockett R W 1983 Asymptotic stability and feedback stabilization. In: Brockett R W, Millman R S, Sussmann H J (Eds) *Differential Geometric Control Theory* Birkhäuser, Boston, MA, pp 181–191
7. Campion G, Bastin G, d'Andréa-Novel B 1996 Structural properties and classification of kinematic and dynamic models of wheeled mobile robots. *IEEE Transactions on Robotics and Automation* 12:47–62
8. Campion G, d'Andréa-Novel B, Bastin G 1991 Modeling and state feedback control of nonholonomic mechanical systems. In: *Proceedings of 30th IEEE Conference on Decision and Control* Brighton, UK, pp 1184–1189
9. Canudas de Wit C, Khennouf H, Samson C, Sordalen O J 1993 Nonlinear control design for mobile robots. In: Zheng Y F (Ed) *Recent Trends in Mobile Robots* World Scientific Publisher, pp 121–156
10. Canudas de Wit C, Sordalen O J 1992 Exponential stabilization of mobile robots with nonholonomic constraints. *IEEE Transactions on Automatic Control* 37:1791–1797
11. d'Andréa-Novel B, Bastin G, Campion G 1995 Control of nonholonomic wheeled mobile robots by state feedback linearization. *International Journal of Robotics Research* 14:543–559
12. De Luca A, Di Benedetto M D 1993 Control of nonholonomic systems via dynamic compensation. *Kybernetika* 29:593–608
13. De Luca A, Oriolo G, Samson C 1998 Feedback control of a nonholonomic car-like robot. In: Laumond J P (Ed) *Robot Motion Planning and Control* Springer, London, UK, pp 171–253

14. De Luca A, Oriolo G, Vendittelli M 2000 Stabilization of the unicycle via dynamic feedback linearization. In: *Preprints of 6th IFAC Symposium on Robot Control* Vienna, A, pp 397–402
15. Fliess M, Lévine J, Martin P, Rouchon P 1995 Design of trajectory stabilizing feedback for driftless flat systems. In: *Proceedings of 3rd European Control Conference* Rome, I, pp 1882–1887
16. Isidori A 1995 *Nonlinear Control Systems* (3rd ed) Springer, London, UK
17. Jiang Z-P, Nijmeijer H 1999 A recursive technique for tracking control of nonholonomic systems in chained form. *IEEE Transactions on Automatic Control* 44:265–279
18. Jones J L, Flynn A M 1993 *Mobile Robots: Inspiration to Implementation* A K Peters, Wellesley, MA
19. Lucibello P, Oriolo G 2001 Robust stabilization via iterative state steering with an application to chained-form systems. *Automatica* 37:71–79
20. Ma Y, Kočeckà J, Sastry S 1999 Vision guided navigation for a nonholonomic mobile robot. *IEEE Transactions on Robotics and Automation* 15:521–536
21. M'Closkey R T, Murray R M 1997 Exponential stabilization of driftless nonlinear control systems using homogeneous feedback. *IEEE Transactions on Automatic Control* 42:614–628
22. Morin P, Samson C 1997 Application of backstepping techniques to time-varying exponential stabilisation of chained systems. *European Journal of Control* 3:15–36
23. Murray R M 1993 Control of nonholonomic systems using chained forms. *Fields Institute Communications* 1:219–245
24. Neimark J I, Fufaev F A 1972 *Dynamics of Nonholonomic Systems* American Mathematical Society, Providence, RI
25. Oriolo G, Panzieri S, Ulivi G 1998 An iterative learning controller for nonholonomic robots. *International Journal of Robotics Research* 17:954–970
26. Samson C 1993 Time-varying feedback stabilization of car-like wheeled mobile robots. *International Journal of Robotics Research* 12:55–64
27. Samson C 1995 Control of chained systems. Application to path following and time-varying point-stabilization of mobile robots. *IEEE Transactions on Automatic Control* 40:64–77
28. Samson C, Ait-Abderrahim K 1991 Feedback control of a nonholonomic wheeled cart in cartesian space. In: *Proceedings of 1991 IEEE International Conference on Robotics and Automation* Sacramento, CA, pp 1136–1141
29. Schraft R D, Schmierer G 1998 *Serviceroboter* Springer, Berlin, D
30. Sepulchre R, Janković M, Kokotović P 1997 *Constructive Nonlinear Control* Springer, London, UK
31. Sontag E D 1990 Feedback stabilization of nonlinear systems. In: Kaashoek M A, van Schuppen J H, Ran A C M (Eds) *Robust Control of Linear Systems and Nonlinear Control* Birkhäuser, Cambridge, MA, pp 61–81
32. Sørдалen O J, Egeland O 1995 Exponential stabilization of nonholonomic chained systems. *IEEE Transactions on Automatic Control* 40:35–49
33. Vendittelli M, Oriolo G 2000 Stabilization of the general two-trailer system. In: *Proceedings of 2000 IEEE International Conference on Robotics and Automation* San Francisco, CA, pp 1817–1823

Landmark Recognition in Indoor Navigation by Fuzzy Maps and CBR

Alessandro Micarelli, Stefano Panzieri, Lorenzo Sciacvico, and Giovanni Ulivi

Dipartimento di Informatica e Automazione

Università degli Studi di Roma Tre

<http://www.dia.uniroma3.it>

Topological graphs can be used as a world representation for a mobile robot that navigates in an office-like environment. Nodes of the graph can represent the intrinsic structure of the environment like corridors, corners and so on. Arcs can capture the connectivity of the space. The task of building algorithms able to identify the characteristic features of nodes directly from sensory data requires several intermediate steps. They consist in the collection of the raw data, their reduction to a possibly one-dimensional representation, its “filtering” and, finally, the “feature recognition”. In the paper several of the many possible methodological choices are shown, with a priority to the (according to the authors) most advanced ones.

In particular, for the “feature recognition” step, two different policies are shown: one founded on the use of a static case library that takes into account the previous results through the Transferable Belief Model, and a second one exploiting an architecture based on Case-Based Reasoning, a method from the Artificial Intelligence domain. Using the latter the robot acquires knowledge on a progressive basis and is therefore able to navigate autonomously in an environment without any prior information.

1 Introduction

Mobile robotics is on the borderline between Automatic Control and Artificial Intelligence. Scientists from the two communities use autonomous robots as a case study for their theories and to assess the validity of some specific techniques. As a matter of fact, both the approaches have their strong points and the problems found in mobile robotics are so wide in scope that everybody can find his own suitable situation. According to a very sketchy comparison, a control theory approach to navigation provides algorithms able to manage directly raw sensor data and to guarantee some nice convergence properties under suitable assumptions. On the other hand, a speculative approach can draw plans to tackle with complex environments and robot interactions. However it needs symbolic information that generally is not available in real time and that must be derived from the measures by classification procedures. The chapter is intended to introduce several useful techniques, to scan some of the

existing literature, and to show in detail an experimentally verified approach that bridges this gap and tries to get the most from the two points of view.

The chapter is based on a two-year long research work in the field; during this period several methodologies have been considered and tested, with the aim of using a more abstract representation of knowledge both in planning and in navigation. The overall navigation structure has been kept rather constant during the research and it is briefly described in Section 3. It is composed by several layers, each one can be implemented taking different approaches.

In the chapter, a review of some advanced possibilities is referred for the layers. First, a short review of the main concepts about topological maps and fuzzy morphology is given. Then, Section 4 describes how to build and enhance a 2-dimensional data structure (a map) based on rough measures. Section 6 shows a data reduction procedure to derive a one-dimensional representation of the space around the robot and, then, Section 8 details different approaches to associate a label to this representation, i.e. to recognise the kind of site where the robot currently is. Final sections refer some experimental results and give indications about future work.

2 World Representation

An effective environment representation for an autonomous mobile robot must describe all the essential information necessary for self-localisation, motion planning and navigation. All those information should be easily extracted from sensory data.

The mapping approaches, i.e. the way the world is represented, that have been proposed in literature can be grouped in two main classes [3]: *metric maps* and *topological maps* (see Figure 2.1). A metric map represents the environment in terms of geometric relations between the objects and a fixed reference frame. On the other hand, in a topological map, only adjacency relations between objects are represented, avoiding metric information as far as possible¹.

Metric maps and topological maps are two different representations of the same environment: as a consequence, they exhibit complementary rather than opposite properties. Metric maps are adequate to represent sensory metric information and are necessary for the implementation of metric based algorithms like the shortest path search. On the other hand, this representation is often space consuming, heavy for long range planning, sensitive to odometric errors, and last, has a poor interface to symbolic system as, e.g. expert systems.

¹ The term “topological” in the title is indeed used in a loose sense to mean “without a metric”.

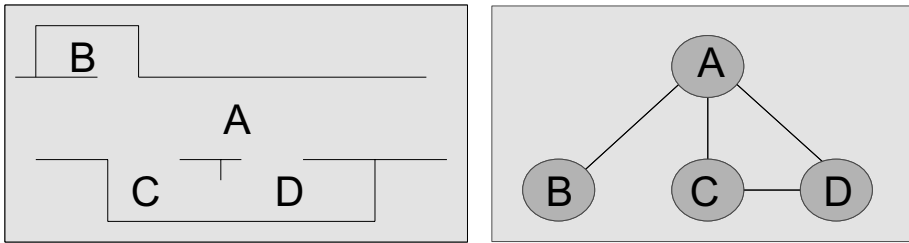


Figure 2.1. Metric (left) vs. topological (right) maps

A topological map is typically a more abstract representation than a metric one and can be synthetically represented by a graph [7], even if the semantic associated to nodes and arcs can differ depending on authors' definitions [14], [25], [10]. This is a compact representation that does not require accurate determination of the robot position for a (topologically) efficient planning. For this reason, and considering that the graph can be enriched with extra-topological information, it can be effectively used for symbolic planning and navigation in an unstructured environment and in particular when considering long displacements. Moreover, they lead to friendly man-machine interfaces as tasks are described (almost) without using metric information by phrases as “follow the corridor and turn right at the first corner”. The main disadvantage of topological maps is that direct extraction of abstract information from sensory data is quite a difficult process and can lead to a *perceptual aliasing*. Finally, precise motion planning is not feasible and, more often, only a suboptimal path can be computed to reach a goal.

To overcome the problem concerning the difficulty of extracting abstract information from sensors, we can exploit the structure of the workspace, especially when the robot navigates in office-like environments: this kind of indoor environment is usually structured with standard elements (places) like corridors, corners and doors.

A graph can easily describe such environments, with nodes representing basic features and arcs symbolising the connections between them. This representation is well suited for long-range navigation: for example, consistent motion strategies can be attached to each node. One example could be a corridor node with a set of strategies that can be explicitly suited for it (follow fast/slowly, follow near the right/left wall, etc.) and can be used to build a navigation sequence. Adding some more attributes can enrich the nodes. One attribute can be an approximate measure, for example the length of the corridor; other attributes can be the number of doors on the left and on the right. Clearly, in order to correctly execute the motion plan, the robot must localise itself on the graph, that is, the robot has to recognise in which node it is.

Therefore, a topological map is defined as a graph (see right side of Figure 2.2), in which nodes correspond to places and arcs connect two nodes if they are adjacent in the real environment. In particular, we consider here only five kinds on nodes: four corresponding to the places *corridor*, *corner*, *end-corridor* and *T-junction*, plus an additional one named *open-space*, that can be useful when the corridor becomes too wide. Each node is augmented with (at least) the motion strategy, so that it can correctly represent together the environment structures and the relevant control strategies.

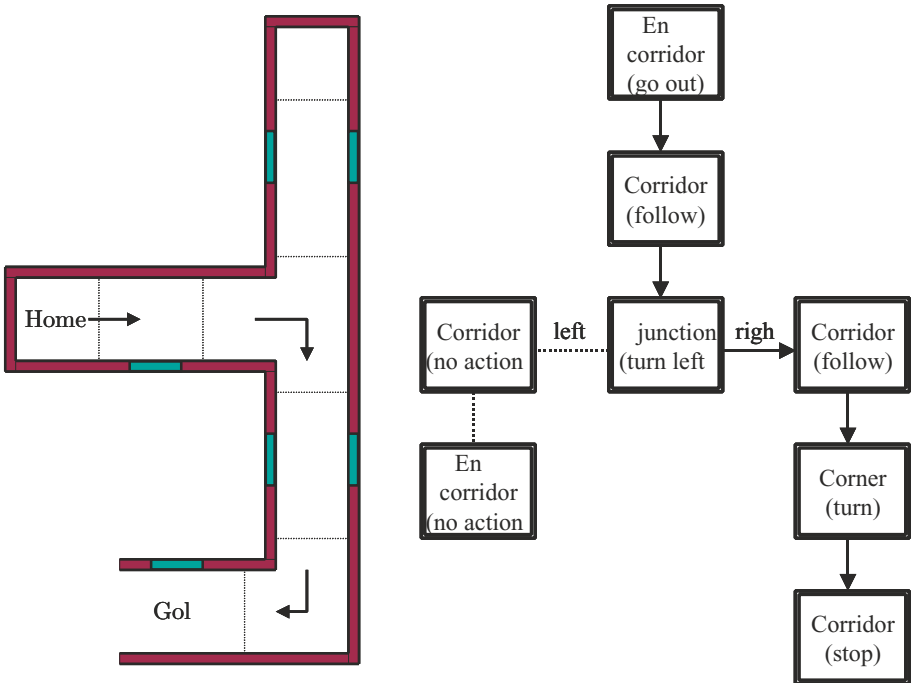


Figure 2.2. Typical office-like environment and its topological representation

3 The Overall Structure

To describe the overall structure adopted in the research, we will follow the data flux, starting from the measures and ending with the decision about the node occupied by the robot.

The main idea consists in comparing an egocentric representation of a “small” area around the robot with some prototypes representing the basic

features considered as the basis for the environment description. The comparison must be efficient and fast and thus noise and unnecessary details should be avoided.

The first step consists in collecting the readings from the ultrasonic sensors as the robot moves, and to build a small fuzzy local map (FLM, as defined in next section) using data from the recent measures. This map generally is quite noisy, as the robot is moving and there is little time to collect many readings. Therefore, a first level of filtering may be useful to improve its quality: algorithms from fuzzy morphology have been found quite effective (see Section 5). The decision about the opportunity of this enhancement depends on the quality of the measures (and therefore on the kind of the environment) and on the available computing power, as it is rather time consuming.

An efficient comparison requires reducing the information from two dimensions to one dimension. We will see that, during this phase, the metric information are discarded and only the structural one is kept. The output of this step will be called a “worldmark”. Again, the data from this step can be improved, in particular if the two-dimensional filtering is skipped. To this aim, an approach based on wavelets theory can be considered that gives very good results. Moreover, it can be used to derive numerical coefficients for the comparison phase.

The comparison between the worldmark and the available prototypes can be effected in several ways. In a previous work [9], the authors proposed a fuzzy similarity measure. It works fine, but has a small, static database of prototypes. A more flexible approach encompasses Case-Based Reasoning, and allows adding new prototypes dynamically to the database by a supervised learning approach. Note that for each feature, there can be more than one prototype according, e.g. to the robot position during the measures. More comparison choices will be discussed, as a suggestion for further research. All the comparison techniques provide one or more matching prototypes sorted by their similarity with the worldmark. Often this is sufficient to decide the actual robot position, but adding information on the previous positions increases, obviously, the robustness of the decision.

4 Fuzzy Maps

A *Fuzzy Local Map* (FLM) can be defined as a *Fuzzy Map* [18] representing the surroundings of the robot. This geometric map is divided into cells, and for each one, two values specifying the degree of membership to the set of empty cells and to the set of occupied cells are given. A FLM is thereafter represented by two fuzzy sets: the empty cells set \mathcal{E} , and the occupied cells set \mathcal{O} . The FLM is derived at each step merging the last n sets of collected data. It represents the world around the robot.

Those fuzzy sets can be employed in several navigation tasks, alone or combined to obtain new fuzzy sets like the one describing the degree of collision danger of navigating into a cell. This set is often referred as *navigation map* (\mathcal{N}) and can be used, for example, to plan fine motions to cross a door, or to go round an obstacle, or to escape from a blind alley. For its description see [18]. Some \mathcal{N} sets obtained from FLMs are reported at left side of Figures 4.1 and 4.2. In this case a FLM of 40 by 40 cells is used, each one covering a space of 10 cm by 10 cm. Note the difference between the white portion of the image corresponding to *unknown* cells and the light gray (near the robot) one related to *very low danger* cells.

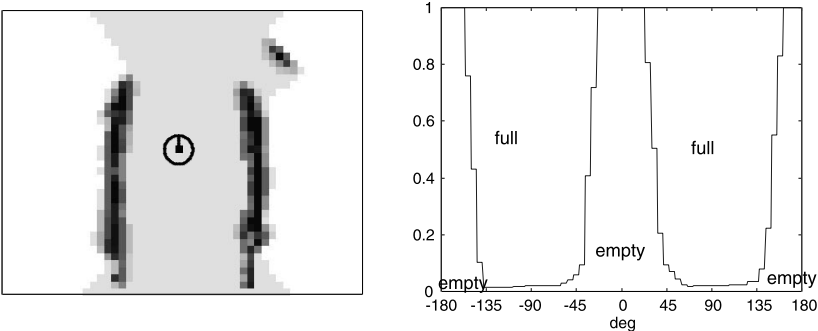


Figure 4.1. Map and World Mark for a corridor

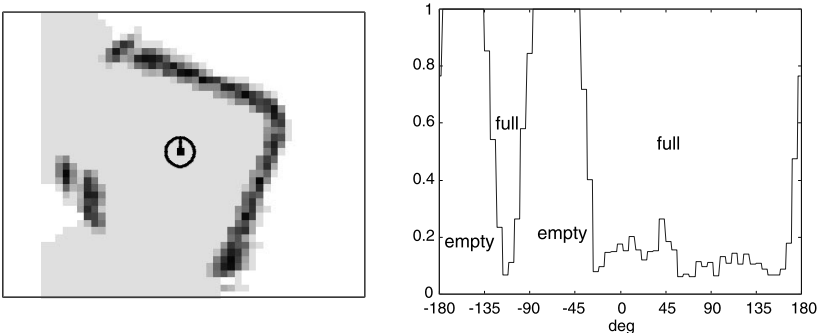


Figure 4.2. Map and World Mark for a corner

The possibility of representing the surrounding of the robot in a metric way using fuzzy logic gives us the ability of fusing information coming from ultrasonic sensors as well as other kinds of exteroceptive sensors (laser scanner, etc.), and opens the way to the use of concepts coming from mathematical morphology.

5 Fuzzy Morphology

Mathematical morphology, based on digital topology theory [19] with its extension to grey-scale images [21] and fuzzy sets [23], is a typical image processing methodology and can be used to extract simple shapes from a digital image and to modify the image itself.

The main idea is to examine the structural content of an image by matching it with a small pattern (“structuring element”) at various locations. Only two elementary operations, called *dilation* and *erosion*, are required to build up complex operators like *closure* and *opening* ones. If the structuring element is a white disk (left of Figure 5.1) the *opening* erases the white holes smaller than the disk. If the structuring element has a conical distribution of grey levels, that can be interpreted as a fuzzy membership function representing “large open space” (right of Figure 5.1), the result after transformation is that the value of each cell will represent how much the cell belongs to the set of cells that are in a “large open space”. This means that very small holes will be definitively closed, but larger ones will be only decreased depending on their size.

Those concepts can be usefully employed in a fuzzy navigation map: for example the shape of poorly mapped walls can be effectively bettered, as shown by Figure 5.2. Here, the application of the opening operator is presented for the image of a corridor with many small holes. They could easily lead to misinterpretation during the comparison phase. The same operator in Figure 5.3 is applied to a T-junction image: the improved one, on the right, has less noise and can be better interpreted as a place with open space back, forth and on the right. The size of the structuring element is critical; indeed it is possible to “close” small holes but also doors along a corridor.

6 Worldmark

To ease the applicability of this approach to a real-time navigation problem, an important simplification is applied by replacing the full 2-D fuzzy description with a polar representation. The \mathcal{N} set (navigation map) is used to build a suitable 360 degrees *horizon* representation of the world around the robot (*worldmark*). The worldmark can be seen again as a fuzzy set, calculated by associating to each direction a degree of emptiness of the space around the

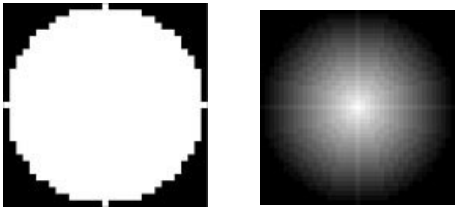


Figure 5.1. Structuring elements for morphological opening

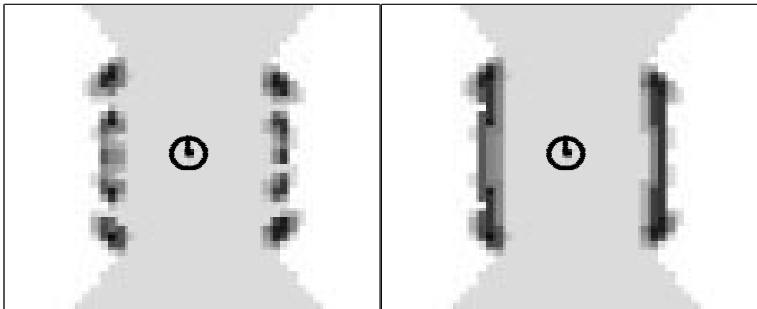


Figure 5.2. Opening on a poor corridor image

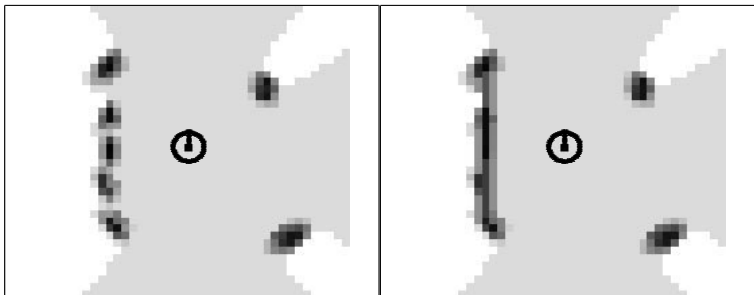


Figure 5.3. Opening on a poor T-junction image

robot. In particular, first the center of gravity of \mathcal{N} is calculated, then 60 rays are taken from there, one every 6 degrees, along which the distance of the maximum value of collision danger (darkness) is taken as a measure of the (inverse of) degree of emptiness along that direction. The center of gravity is used to reduce the sensitivity to the actual robot position and to improve the representation of the spatial distribution of walls and empty spaces in the image.

As the distances from the center of gravity are not used in the worldmark computation, this actually carries only structural information about the space around the robot.

Two examples of worldmarks are reported at the right side of Figures 4.1 and 4.2: note that “full” labels associated to each direction mean that there is a wall, while an “empty” label refers to the presence of a free space (in the short distance). As it can be seen in Figure 4.2 the angular shape of the corner is not caught in this simplified analysis while it is correctly understood that there are two free spaces in orthogonal directions. This can be enough for the aim of navigating in office-like environments but pushes for a more accurate investigation to extend the accuracy of this technique.

Now, the general comparison problem is reduced to matching this *worldmark* with p prototypes from a static or dynamic library. This matter will be developed in Section 8.

7 Wavelet Decomposition

Wavelet Transform theory is a rather recent concept, since the first references in the literature date back about 10 years [15], [11]. Their success is linked to their specific property whereby they may be located both in time (space), as well as in scale (frequency), thus providing a “time-scale map” of the signal, wherefrom it is possible to extract time varying features. Due to this special property wavelets are an ideal tool for the analysis of physical situations characterised by signals of a discontinuous nature and featuring sharp variations, as is our case. The analysis procedure, by means of wavelets, is based on the use of a prototype function, so-called “mother wavelet”, whose translated and extended (or compressed) versions constitute the basis functions for the series expansion that allows for the representation of the original signal through coefficients. Data operations may therefore be effected on the corresponding wavelet coefficients. In particular, if the mother wavelet is appropriately chosen, or if the coefficients under a certain threshold are eliminated, it is possible to “sparsely” represent the original data. This means that wavelets are a most useful tool in the context of data compression. The practical, fast computation of wavelet transform has become possible in applications, owing to the fast DWT, also known as Pyramid Algorithm [15]. This algorithm is $O(n)$, where n is the length of the array of data to be transformed; this is

extremely advantageous compared to other algorithms (for example the FFT is $O(n \log n)$).

7.1 Denoising with Wavelet

The above-mentioned considerations explain why wavelets found so many applications, in particular when dealing with noisy signals. To resolve very noisy problems different techniques have been proposed [5], [6], based on the following idea: the decomposition of a data set by means of wavelets involves filters that output both “averages” and “details”; if the latter appear of minor significance, they can be attenuated or eliminated (“wavelet shrinkage or thresholding”) without causing a substantial alteration of the main features of our given data set. This method is exemplified in the diagram shown in Figure 7.1.



Figure 7.1. Data Analysis by Wavelets

Note that the signal is initially transformed, its coefficients are matched against a predefined threshold, and subsequently reutilised to reconstruct the original data set through the inverse DWT. Donoho e Johnstone [5], [6] propose a number of different denoising strategies, using predefined thresholds as well as adaptive ones. Actually, there is no one strategy that always gives the best results compared to the others, but only strategies whose validity depends on the type of application. Once the threshold is determined, it may then be matched against the wavelet coefficients using two different possible methods (see also [16], [17]):

- *Hard Thresholding*, which implies the “keep or kill” principle, i.e. all the wavelet coefficients are checked at an absolute value against a predetermined threshold t_h : if the magnitude of the coefficient is less than t_h , the coefficient is replaced by zero, otherwise it is left unchanged:

$$c_{jk} = \begin{cases} 0, & c_{jk} < t_h \\ c_{jk}, & c_{jk} \geq t_h. \end{cases}$$

- *Soft Thresholding*, in this case, the wavelet coefficients are shrunked towards the origin in accordance to the following expression:

$$c_{jk} = \text{sign}(c_{jk})(|c_{jk}| - t_s)_+.$$

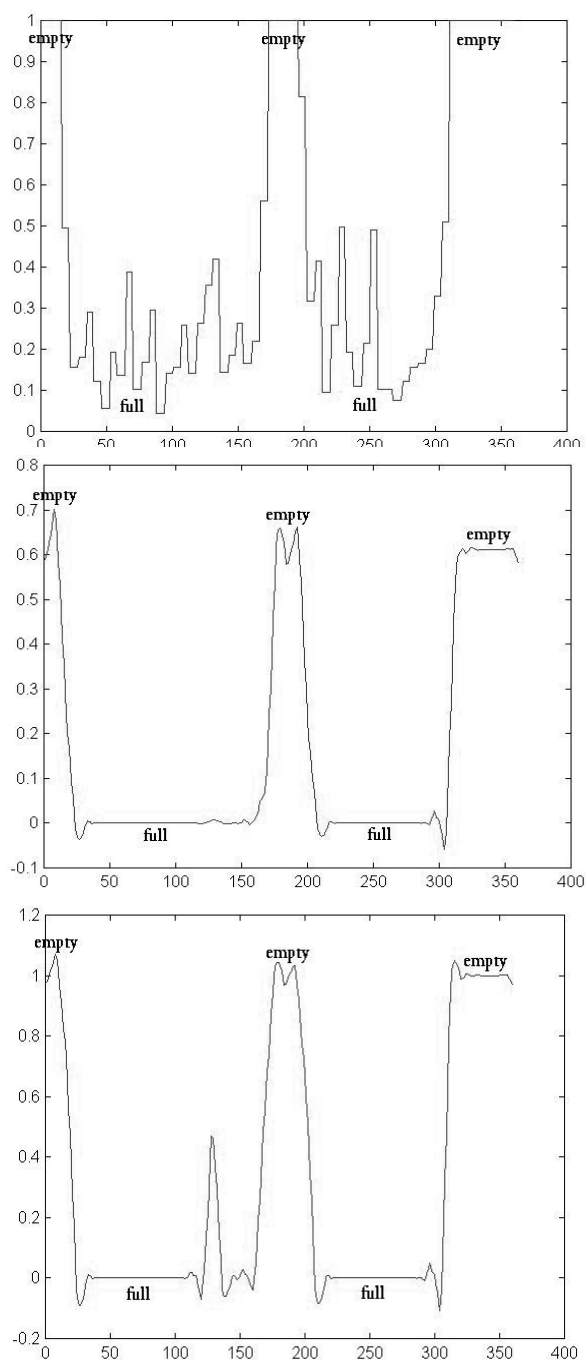


Figure 7.2. Worldmark (top) and reconstructed functions after Soft Thresholding (middle) and Hard Thresholding (bottom)

Figure 7.2 reports the result of the application of the two different methods to the signal shown at the top, using the same threshold value ($t_h = t_s = 1.1$) in both cases (the 3-scales forward and inverse DWT are performed by using Symmlet with 8 vanishing moments as the mother wavelet).

8 Similarity Metric

In this section some techniques that have been extensively tested on the available data will be analysed. The problem is to define a similarity function between the current worldmark and the reference ones. Till now we have defined five kind of places, drawn in Figure 8.1 together with their worldmarks.

Such distance function must satisfy the following four basic properties:

$$\begin{aligned}\Delta(x, x) &= 0 \\ \Delta(x, y) &> 0, \quad x \neq y \\ \Delta(x, y) &= \Delta(y, x) \\ \Delta(x, y) &\leq \Delta(x, z) + \Delta(z, y).\end{aligned}\tag{8.1}$$

In a real-time context, a good metric must guarantee an efficient compromise between two main requirements: quality and computational complexity. To this aim, it will be shown how the effectiveness of the metric will depend on the preprocessing phase, i.e. on the ability of removing extra information leaving only the part functional to the recognition purpose.

8.1 Cross Correlation Similarity

In the context of the experimentation, we adopted different approaches, each of which yielded its specific merits and shortfalls. Among these, the best overall results were given by the Cross-Correlation Factor metric, expressed as follows:

$$\frac{\max(\varepsilon_{xy}(\tau))}{\sqrt{(\varepsilon_{xx}(0)\varepsilon_{yy}(0))}}.\tag{8.2}$$

This quantity was calculated both in the time domain and in the Fourier domain, obtaining important results, with reasonable computational lengths using calculation resources available on the market. Furthermore, this method allowed us to compute the other information available in the mono-dimensional map, that is, the polar information.

8.2 Fuzzy Similarity

Another approach followed, based on the use of fuzzy sets theory [12], consists in computing the degree of matching between the measured worldmarks

and the reference ones, interpreted as fuzzy sets. The comparison was performed by using one out of the many fuzzy similarity functions defined in literature. In our case the following fuzzy similarity function defined by Sokal and Michener, reported in [2], gave reliable results:

$$\text{sim}(\mathcal{M}, \mathcal{L}) = \frac{\text{card}(\mathcal{M} \cap \mathcal{L}) + \text{card}(\overline{\mathcal{M}} \cap \overline{\mathcal{L}})}{\text{card}(\overline{\mathcal{M}} \cap \mathcal{L}) + \text{card}(\mathcal{M} \cap \overline{\mathcal{L}}) + \text{card}(\mathcal{U})} \quad (8.3)$$

where \mathcal{M} is the fuzzy set representing the world mark, \mathcal{L} is a reference set, and \mathcal{U} is the universe of discourse. In particular, as intersection operator, the minimum function has been used and the cardinality is defined as the integral of the membership function.

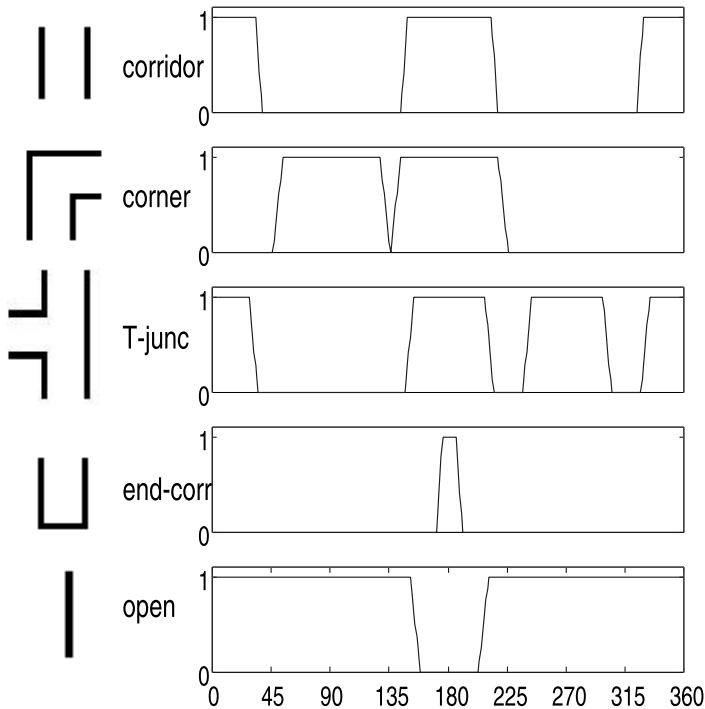


Figure 8.1. Reference sets

To take into account that the relative orientation between the robot and the real feature is not *a priori* known several comparisons have been per-

formed shifting \mathcal{M} w.r.t. \mathcal{L} to cover the possible orientation range. The final resemblance $\text{Res}(\mathcal{L})$ to a feature is computed as the maximum value of all similarities obtained during this process. This step is concluded with the delivery of the p resemblance measures that are in the interval $[0, 1]$.

8.3 Wavelet Similarity

Using the concepts of thresholding and of DWT, it is possible to define a new method to compute a similarity function based of wavelet decomposition.

First, a complexity reduction of the original signal (the worldmark) is performed extracting only few significant coefficients from the DWT, and second, comparing those coefficients with the ones stored for the ideal features in the library. In particular, the forward 3-scale DWT (with Symmlet and 8 vanishing moments) is computed and of the 360 obtained coefficients, only 45, coming from the low-pass filter of the third scale, are retained. Comparisons, using those coefficients, can be made in several ways: the one here adopted is again a normalised cross-correlation. Experimental results, comparing the time required for this cross-correlation (including that to compute the DWT) with the one made on the whole worldmark, give a reduction of more then $1/4$ with no difference for what concerning the feature recognised.

Possible improvements are in the denoising phase that envisages the application of the forward and inverse DWT. Therefore, an interesting objective is to optimise the identification of the mother wavelet and the number of scales to which the wavelet transform is applied. For now, this choice is pre-determined, but it should be possible to develop a model where this happens dynamically, on the basis of feedback on the obtained results.

In the next section we will see how the possibility of reducing the amount of data stored for each worldmark open the way to the use of a larger library, possibly to build starting from data collected during navigation.

9 Extension to a Dynamic Library: CBR

As stated before, the first recognition phase is done in most cases comparing the actual view with a static list of models obtained with *a priori* considerations on the environment itself [9]. Instead, following a CBR philosophy, a learning approach can be devised in which several real-world cases are obtained from a supervised navigation and used to build a dynamic library [13], [1]. In this section we want to show how such a method can be successfully applied to this problem in dynamic environments containing features that only partially correspond to the already-known cases.

Assume to have a new worldmark, see Figure 9.1, to be classified. The retrieval module shown in the figure will search in the case library containing the old cases, with a $\langle \textit{problem representation}, \textit{solution} \rangle$ structure, which in

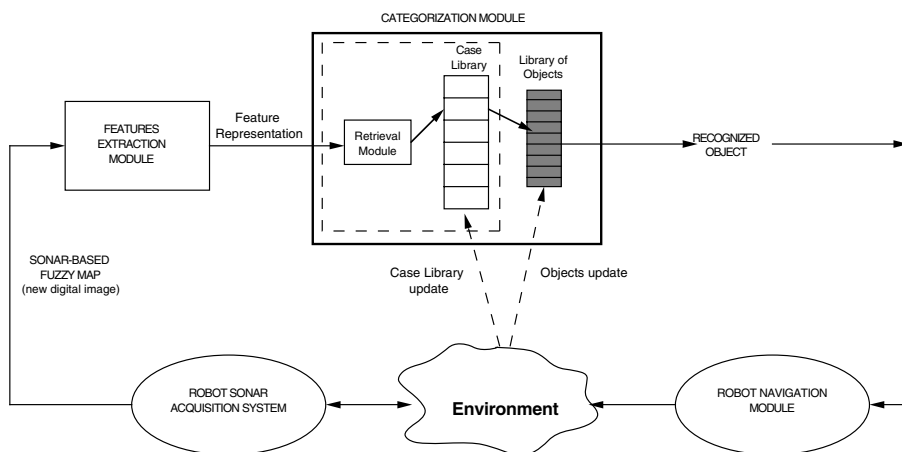


Figure 9.1. Case-Based Approach to Image Recognition for Indoor Navigation

this specific case will be $\langle \text{worldmark}, \text{topological feature represented by the image} \rangle$. The solution given in the old case can therefore be seen as a pointer to a new library, which can be named “Library of Objects”, containing the categories (i.e. “topological features”) that could be present in the worldmark to be analysed. The “Recognised Feature” is at this point taken into consideration by the robot navigation system to plan its motion. This object, which constitutes the old solution of the case retrieved from the Case Library, will also be considered as the solution of the new problem (basically, there is no need for an adaptation of the old solution to suit the new case) and if the supervisor accepts it, the pair $\langle \text{New Worldmark}, \text{Recognised Feature} \rangle$ can be inserted as a new case in the Case Library.



Figure 9.2. Image Recognition Module

Let us now analyse further in detail the architecture of the module shown in the Figure 9.2. For sake of clarity, we provide a simplified version to illustrate the problems addressed and the solutions proposed. The algorithm

has been implemented in C since the entire control procedure in respect to the robot's navigation was originally developed using this language. For a better understanding of the procedure, the algorithm reported in Figure 9.3 is based on the use of simplified data structures as opposed to the ones actually implemented. We assume the use of a record consisting of two fields for the case representation, both as concerns the new case as well as any other in the Case Library. The first field is reserved to the representation of the image. Therefore, the "new digital image" practically consists of an array of 360 real numbers between 0 and 1 (the worldmark). Instead, the second field is reserved to the recognised object, where "object" is intended as an integer corresponding to one of the above possible categories. Accordingly, the set of instructions given in pseudo-code is as described in Figure 9.3.

This structure envisages the possibility of intervention by a human expert, who is charged with the initial training phase of the robot as well as the checking phase of the retrieved solution. As shown, the use of two different threshold values S_a and S_b are postulated and the former controls whether the case solution retrieved from the library is, or is not, extended to the new case. Basically, S_a is the "reliability threshold" that rates the possibility for the case extracted from the library to be considered as representative of the one under observation. Instead, the possibility of the insertion of the new case in the Case Library is dependent on S_b , the so-called "identity threshold". The reason for the existence of this second threshold is obvious. Indeed, a richer case library increases the possibility of retrieving the best matching case to the input one and, consequently, improves the accuracy of matching the found solution to the new situation.

On the other hand, an extensive library involves two drawbacks:

- more time is required to retrieve cases from memory;
- more memory capacity is needed.

In fact, if we imagine the Case Library as an array of records, each time a case is input to our module, the procedure is run through from the beginning to the end, therefore the computational complexity of this operation is linear to the number of available cases. As a matter of fact, these are actually underlying problems common to all CBR applications, as also demonstrated by the considerable size of material generated by the Artificial Intelligence community on these concerns. Various solutions [20] have been proposed, which can be grouped into three categories:

- flat memory;
- discriminant nets;
- E-MOP (Episodic Memory Organisation Packets).

On the basis of the obtained results, we can affirm that the computational burden ensuing from a denoising operation appears compensated by the significant reduction of cases to be inserted in the library. This is due to

```

Function REC(NewImage) returns RecObject
  inputs : NewImage; the input image
  variables : CaseLib; the case library
                $C_j$ ; the generic old case
                $S_a$ ; the reliability threshold
                $S_b$ ; the identity threshold
  local variables : D.image; the image representation
                     D.object; the recognised object
                      $s_j$ ; the metric value
                     tempvalue; the temporary metric value
                     tempind; the temporary case index

  D.image  $\leftarrow$  DENOISE(NewImage)
  D.object  $\leftarrow$  0
  tempvalue  $\leftarrow$  0
  tempind  $\leftarrow$  0
  for each old case  $C_j$  in CaseLib do
    begin
       $s_j \leftarrow$  COMPARE_CASE(D.image,  $C_j$ .image )
      if (tempvalue <  $s_j$ ) then
        begin
          tempvalue  $\leftarrow$   $s_j$ 
          tempind  $\leftarrow$   $j$ 
        end
      end
      if (tempvalue <  $S_a$ ) then
        begin
          D.object  $\leftarrow$  HumanExpertSolution
           $C_{n+1}$ .image  $\leftarrow$  D.image
           $C_{n+1}$ .object  $\leftarrow$  D.object
        end
      else
        begin
          if ( $C_{tempind}$ .object = HumanExpertSolution) then
            D.object  $\leftarrow$   $C_{tempind}$ .object
          else
            D.object  $\leftarrow$  HumanExpertSolution
          if (tempvalue <  $S_b$ ) then
            begin
               $C_{n+1}$ .image  $\leftarrow$  D.image
               $C_{n+1}$ .object  $\leftarrow$  D.object
            end
          end
        end
      RecObject  $\leftarrow$  D.object
    returns RecObject
  
```

Figure 9.3. The Image Recognition Algorithm

the fact that by eliminating the overlaying noise the similarity of two situations corresponding to the same topological condition may often be revealed, while their respective representations covered by noise misleadingly appear considerably different. Moreover, we have already seen how the computation length and the quantity of utilised resources are directly proportional to the number of cases kept in memory.

10 Combined Recognition

The first step of the recognition problem ends with the formulation of the “best” feature matching the actual sonar view. This cannot satisfy all the needs for a correct navigation which, in principle, should be prepared at different source of noise: at least people moving around the robot and giving it a wrong “impression” of the environment. This advocate for a method able to fuse data and information of different nature and coming from past navigation history such as external agents.

Thereafter, for the second step of the recognition process, an algorithm can be devised on the base of the *Transferable Belief Model* theory (TBM) [4], [22], [24]. This method is based on the use of the similarity function adopted in the first recognition phase as an agent that assigns a degree of belief on each feature when new data become available. Those beliefs, to add robustness with respect to disturbances coming from multiple reflections or unexpected obstacles (e.g, furniture, walking people, etc.), are combined with past information or any other belief coming from other independent agents. For example, if the node was labeled with a length l , this information could be used to force an opinion that the corridor is reaching its end as far as the robot is estimating that the path traveled is approaching l .

Let $X = \{C, R, T, E, O\}$ be the universal set containing the corridor C , the corner R , the T-junction T , the end-corridor E , and the open-space O . Let x be an element of X , and the event F an element of $P(X)$, the power set of X . All the possible events Ω include the empty set (contradiction) and the universal set (unknown).

At each step, a *basic mass assignment* (BMA) $m_r(F)$ is determined: this function quantifies how much the event F is supported by the available information. In our case, considering the fuzzy approach to the evaluation of similarities, the available information consists of the resemblance values calculated in the static recognition. A possible BMA can be obtained supposing that each resemblance $\text{Res}(\mathcal{L})$ supports with its value the atomic event L associated to the reference set \mathcal{L} , and with its negate both the complementary and the unknown events:

$$m_r(F) = \begin{cases} \text{Res}(L)/p & \forall F \text{ atomic with } F = L \\ (1 - \text{Res}(L))/2p & \text{if } F = \overline{L} \\ (\sum_L (1 - \text{Res}(L)))/2p & \text{if } F = X. \end{cases} \quad (10.1)$$

This mass distribution guarantees that BMAs add to one.

To update the past BMA $m_{k-1}(F)$ and combine it with the new BMA $m_r(F)$ the following rule of combination is used:

$$m_k(F) = \sum_{G \cap H = F} m_r(G) \quad m_{k-1}(H). \quad (10.2)$$

Now, following [24], the probability function BetP, used to make decisions, can be written as:

$$\text{BetP}_k(F) = \sum_{G: F \in G \subseteq \Omega} \frac{m_k(G)}{\text{card}(G)}. \quad (10.3)$$

Note that $\text{BetP}(\cdot)$ is computed only for the atomic events F : comparing those values the best hypothesis and its level of belief can be determined.

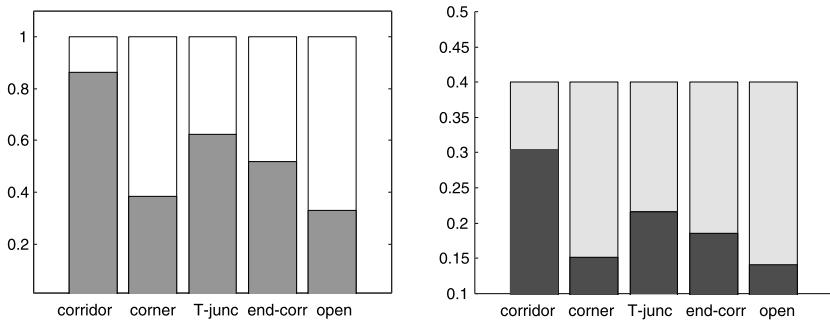


Figure 10.1. Resemblance and Bet for the corridor

To give some examples, consider Figures 10.1 and 10.2: they show the resemblance and bet values relative to Figures 4.1 and 4.2 respectively. Note the difference between resemblance, that are related to one reading only, and bet values, taking into account the history due to past readings: e.g. in Figure 10.1, the final degree of belief in the corner hypothesis is lower than the corresponding resemblance value due to the conflicts between this hypothesis and both other ones and past history. Moreover, in case of erroneous readings, the method is able to filter out incorrect resemblance.

An important role plays the mass of the empty set. Remember that this value increases every time some contradiction is produced by resemblance functions. So, as far as the robot navigates and changes its world mark, its destiny is to collect all the masses distributed among the possible events and to reach the unitary value. This would imply a freezing of the bet values to a zero probability. Now, to allow this *changing of opinion*, at every iteration a

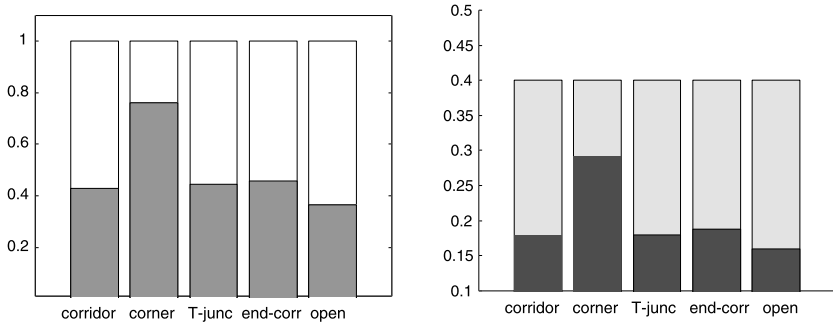


Figure 10.2. Resemblance and Bet for the corner

small percent of the empty set mass (10%) is reversed into the universal set and can be used, through the TBM, to feed new opinions. Finally, the mass of the empty set can be also used to judge the quality of the bet values: when increasing, the level of contradiction of measures is high and an extra analysis could be performed, for example slowing the robot and allowing more sonar readings.

Note that the choice on how distribute the BMA could be changed if, for example, we would like to reward the winner (the one with the maximum similarity) and all the subsets containing it and the other atomic elements. As opposed to the previous choice, we never attribute a mass to the atomic elements that are not winning the “competition”. Looking at Figure 10.3 we see how masses are distributed only to the sets $\{C\}$, $\{C, E\}$, $\{C, T, E\}$, $\{C, T, E, O\}$, and $\{C, R, T, E, O\}$ (unknown). In this case, results obtained show the system less incline to change its opinion when a new feature is approaching due to a stronger bet value, but more firm in devising the right one in presence of some noise.

From experimental tests, we found that the proposed recognition algorithm is computationally efficient and, thus, can be performed in real time during navigation. The complete cycle, including acquisition, FLM building, feature recognition, and motion control with an obstacle avoidance algorithm, takes about 0.5 s on a Pentium I at 133 MHz running Linux.

11 Experiments with Static Library and TBM

The method has been tested on a Nomad 200 equipped with a ring of 16 equally spaced ultrasonic sensors. The environment used for the experiments is a branch of our department, schematically depicted at left-side of Figure 2.2. On the right side of the same figure is shown the corresponding

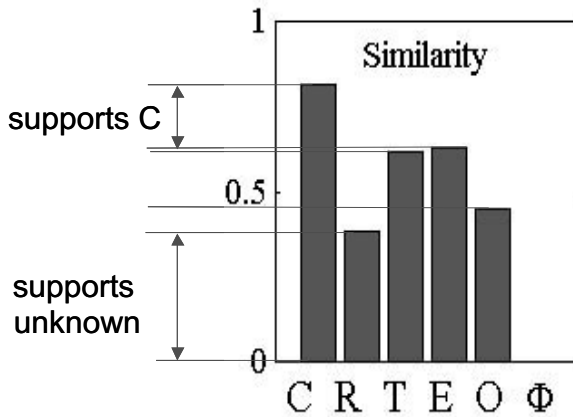


Figure 10.3. Alternative Basic Mass Assignment

graph that can be easily built by the user. The oriented arcs in Figure 2.2 represent a predefined navigation plan: the label on the node inside the round brackets specifies the motion that the robot has to execute when it is in that node.

To demonstrate the validity of the proposed approach, let us consider a sequence of three steps near the goal, shown in Figure 11.1. On the left side, we can see the local fuzzy maps built from ultrasonic measures collected in the last three steps. This is done to have more information about the surrounding space and to filter out spurious reflections. On the right, the beliefs on each feature are represented.

Robot in the corridor: the robot sees two large empty spaces in the front and in the rear. Resemblances give for many steps a large contribution to the mass of the corridor and the belief on this feature is much higher than on the other ones. The robot is navigating into the topological map and is performing a *follow corridor* behaviour.

Robot between the corridor and the corner: now beliefs are almost the same because of the ambiguity of the sensory data. The empty set mass (not reported) is increasing so the robot should be slowed down. Actually, this is automatically done by a lower level safety module that is not described in this chapter, an obstacle avoidance module [8] that forces the robot to move only inside a safety area and decreases robot speed when obstacles are too near.

Robot in the corner: the new feature has been correctly and repeatedly identified so its bet is now sensibly higher than others. Robot position inside the topological map is updated and a new behaviour will be executed to turn the robot left and step into the next corridor. The latter will be recognised very soon.

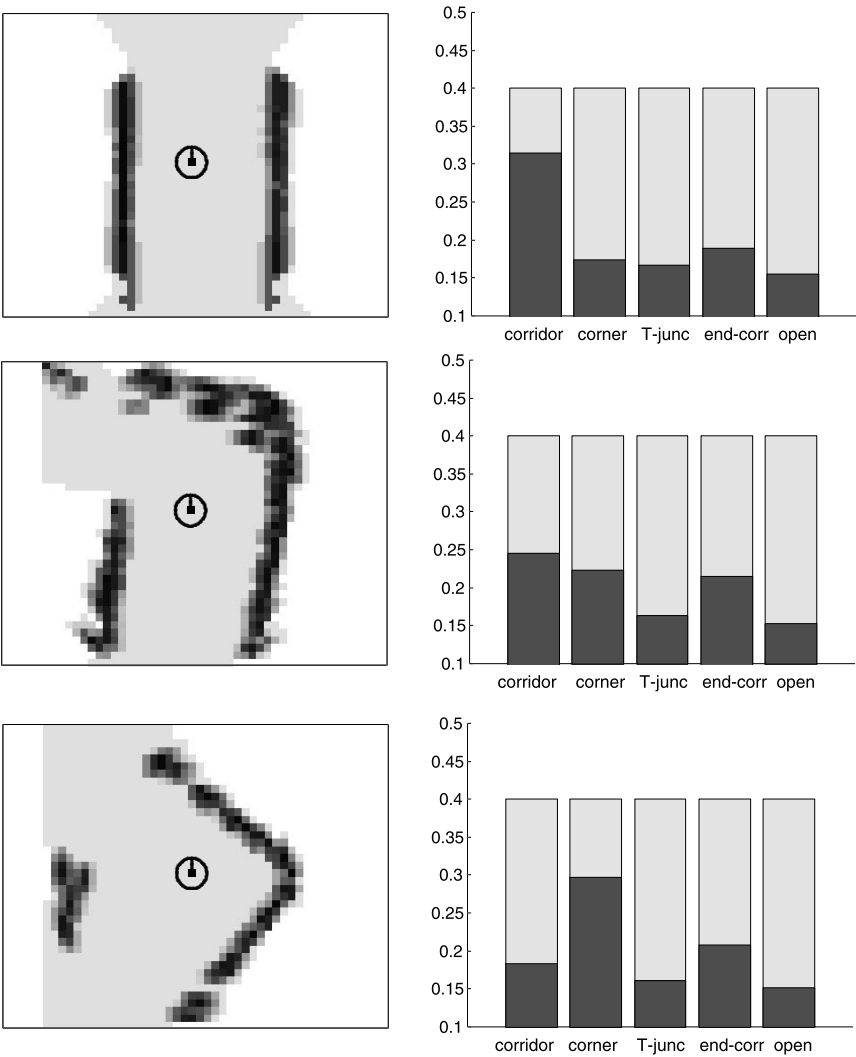


Figure 11.1. Transition from a corridor to a corner

12 Conclusion and Future Developments

This chapter has presented a structure for indoor landmark recognition. It is composed by several layers. Using this structure, several techniques that can be usefully adopted have been described, in particular as it concerns the “highest” layers.

An interesting proposal is to use CBR to compare the acquired data with a dynamic library of prototypes.

Indeed, the normal pattern recognition techniques require models of the objects that must be recognised and classified. The collection of models available to the classifier clearly reflects the original comprehension of the situation to be analysed. Sometimes an *a priori* knowledge can be formalised and used to build such models; however, in most cases, as for the robot’s autonomous navigation, there exists practically no prior information whatsoever. If a traditional feature extraction algorithms is incorporated into a CBR system, a constant increase in the knowledge of the surrounding environment is possible. Furthermore, it should not be forgotten that there is also the possibility of updating the Object Library as well as the Case Library.

Finally, a second step can be taken to improve the navigation algorithm. To fuse past and actual believes and merge approximate metric information with a similarity metric, in view of a correct deduction of a final belief about which feature is more plausible, a Dempster-Shafer approach can be successfully applied.

References

1. Aamodt A, Plaza E 1994 Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Communications* 7(1):39-59
2. Bandemer H, Näther W 1992 *Fuzzy Data Analysis* Kluwer, London, UK
3. Borenstein J, Everett H R, Feng L 1996 *Navigating Mobile Robot: Sensors and Techniques* Peters A K, Wellesley, MA
4. Dempster A P 1968 A generalization of Bayesian inference. *Journal of Royal Statistical Society: Series B* 30:205-247
5. Donoho D 1992 Denoising via soft-thresholding. *IEEE Transactions on Information Theory* 41:613-627
6. Donoho D, Johnstone I M 1995 Adapting to unknown smoothness via wavelet shrinkage. *Journal of American Statistical Association* 90:1200-1224
7. Dudeck G, Jenkin M, Milios E, Wilkes D 1991 Robotic exploration as graph construction. *IEEE Transactions on Robotics and Automation* 7:859-865
8. Fabrizi E, Panzieri S, Ulivi G 1999 An integrated sensing-guidance system for a robotized wheelchair. In: *Proceedings of 14th IFAC World Congress* Beijing, China, pp B:457-462
9. Fabrizi E, Panzieri S, Ulivi G 2000 Extracting topological features of indoor environment from sonar-based fuzzy maps. In: Pagello E, Groen F, Arai T, Dillmann R, Stentz A (Eds) *Intelligent Autonomous Systems 6* IOS Press, Amsterdam, pp 596-603

10. Fabrizio E, Saffiotti A 2000 Extracting topology-based maps from gridmaps. In: *Proceedings of 2000 IEEE International Conference on Robotics and Automation* San Francisco, CA, pp 2972–2978
11. Jacovitti G, Neri A 1998 Multiscale circular harmonic wavelets: A tool for optimum scale-orientation independent pattern recognition. *Proceedings of SPIE* 3391:348–357
12. Klir G J, Folger T A 1988 *Fuzzy Sets, Uncertainty and Information* Prentice-Hall, Englewood Cliffs, NJ
13. Kolodner J 1993 *Case-Based Reasoning* Morgan Kaufmann, San Mateo, CA
14. Kuipers B, Byun Y T 1991 A robot exploration and mapping strategy based on semantic hierarchy of spatial representation. *Journal of Robotics and Autonomous Systems* 8:47–63
15. Mallat S G 1989 A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 11:674–693
16. Micarelli A, Neri A, Panzieri S, Sansonetti G 2000 A case-based approach to indoor navigation using sonar maps. In: *Preprints of 6th IFAC Symposium on Robot Control*, Vienna, A, pp 345–350
17. Micarelli A, Neri A, Sansonetti G 2000 A case-based approach to image recognition. In: Blanzieri E, Portinale L (Eds) *Advanced in Case-Based Reasoning* Springer, Berlin, D, pp 443–454
18. Oriolo G, Vendittelli M, Ulivi G 1998 Real-time map building and navigation for autonomous robots in unknown environments. *IEEE Transactions on System Men and Cybernetics – Part B: Cybernetics* 28:316–333
19. Rosenfeld A 1979 Fuzzy digital topology. *Information Control* 40(1):76–87
20. Schank R 1982 *Dynamic Memory: A Theory of Learning in Computers and People* Cambridge University Press, New York, NY
21. Serra J 1982 *Image Analysis and Mathematical Morphology* Academic Press, London, UK
22. Shafer G 1976 *A Mathematical Theory of Evidence* Princeton University Press, Princeton, NJ
23. Sinha D, Dougherty E R 1992 Fuzzy mathematical morphology. *Journal of Visual Communication and Image Representation* 3:286–302
24. Smets P, Kennes R 1994 The transferable belief model. *Artificial Intelligence* 66:191–234
25. Thrun S 1999 Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence* 1:21–71

Sensor Fusion for Robot Localisation

Fabrizio Grandoni¹, Agostino Martinelli¹, Francesco Martinelli¹,
Salvatore Nicosia¹, and Paolo Valigi²

¹ Dipartimento di Informatica Sistemi e Produzione
Università degli Studi di Roma “Tor Vergata”
<http://www.disp.uniroma2.it>

² Dipartimento di Ingegneria Elettronica e dell’Informazione
Università degli Studi di Perugia

In this chapter the localisation problem for mobile robots is addressed. The localisation is accomplished using data coming from different sensors, which must be properly combined. For this reason, as a preliminary material, the first two sections of the chapter will be devoted to sensor integration and fusion. Different approaches will be considered according to the specific scenario where the localisation must be executed. The localisation section will present some basic methods and problems and will also report some experimental results. The chapter ends with a conclusions section which summarises the subject and also provides some directions for future research.

1 Introduction

Let us consider the problem of implementing a control system that enables a robot to autonomously execute a given task in a given environment. Let us call *world* the set of robot and environment. The robot control system receives information about the world through robot sensors and modifies the world through robot actuators. This is illustrated in Figure 1.1.

An architecture provides a set of principles to organise a control system. There are four basic families of control architectures:

1. planner-based,
2. reactive,
3. hybrid,
4. behaviour-based.

Traditional Artificial Intelligence (AI) systems are planner-based systems. The system is decomposed in a set of functional modules, as domain independent as possible to increase module reuse. The module interaction is mediated by an internal central world model. The control structure is sequential: each module takes its turn to interact with the internal representation of the world. In general, sensory information is fused by a perception module to update the internal model. Then a reasoning engine, the planner, works on the internal model to generate a sequence of actions. The internal model

should be as complex as the external world, and this fact makes planner-based systems not workable for complex environments, in particular in the dynamic case.

A reactive system consists of a set of condition-action pairs (if necessary, there can be a short-term memory): this system reacts to each “stimulus” with a pre-programmed action. It is important to have a fast feedback from the environment to ensure that the appropriate action is executed. A set of very simple reactions can determine a complex behaviour. For complex environments, it may be very difficult to associate an action to each input.

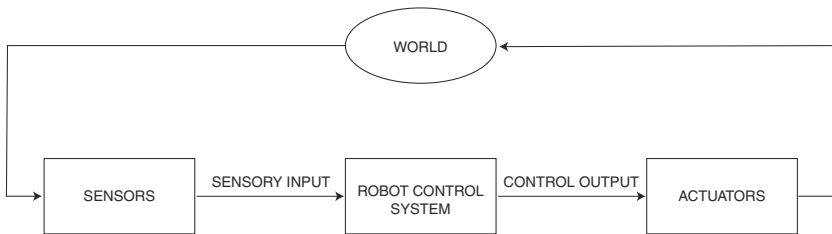


Figure 1.1. Multisensor integration

Hybrid systems combine the potentiality of reactive and planner-based systems. Usually a reactive system handles some low-level real-time tasks (for example to avoid dynamic obstacles) while a planner-based system handles higher-level tasks. As the reactive module can change the world model or the robot state, the planner must be able to re-plan consequently.

Behaviour-based systems consist of task-oriented modules, working in parallel and independently producing commands in response to their particular view of the world (computation and representation are distributed contrary to planner-based systems). An arbitration method is necessary to select or fuse the different commands. The modules communicate directly with very short messages or indirectly through the environment (that is used as a sort of “external model”). A coherent behaviour emerges from modules interaction with the environment. In Figure 1.2 a comparison between a behaviour-based architecture and a planner-based one is showed. Hierarchical structures like those in Figure 1.2 have been also employed for structured environment representation maps in [6].

The attention will be focused on the first kind of architecture and in particular on sensor interpretation problem. The main objective is the determination of the robot position in the environment (*localisation problem*): in order to do it, sensory data must be processed according to some interpretation rule. The precision and accuracy in the position estimation, the

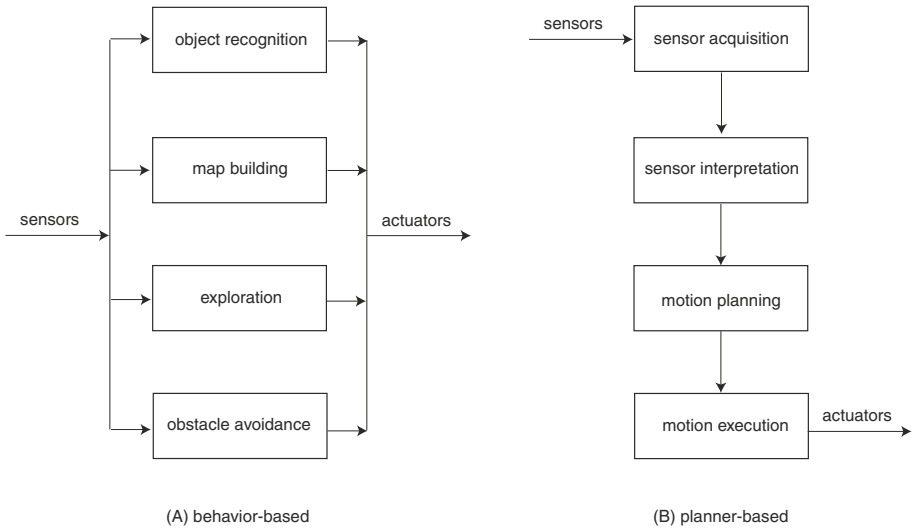


Figure 1.2. Examples of behaviour-based and planner-based architectures

robustness of the localisation method and the reliability of the system are basic requirements in robot localisation. The use of a unique sensor can hardly satisfy these requirements. First, a high precision sensor is much more expensive than many low precision sensors. In addition, the use of a unique sensor can seriously impair the robustness and the reliability of the system. Requirements like precision, robustness and reliability can be greatly improved using a multi-sensor system.

However, data coming from different sensors must be properly integrated. The concept of sensor integration generically denotes the combined use of different types of sensors. Sensor fusion is a more specific concept, which indicates that data from different sensors are combined into one representational format. Redundancy, complementarity, timeliness, uncertainty reduction, reliability are the advantages offered by the use of sensor fusion. Sensor integration will be the subject of Section 2, where the concept of sensor closeness (that is data consistency among different sensors) will be also presented. Section 3 presents the specific topic of sensor fusion.

A different approach is required if the environment statistics are known or unknown. This will be discussed below together with centralised and decentralised architecture implementation.

A typical application of sensor fusion is the mobile robot localisation, which will be presented in Section 4 and is the main subject of this chapter. A Kalman Filter can be used to combine data from the sensors and to localise the robot if an environment representation (let us call it *map*) and a sensor-

environment interaction model are available. If the map is not available, the localisation process and the map construction can be carried out jointly as will be discussed below. More complex localisation problems, related in particular to space exploration missions, concern the localisation of a team of mobile robots. Experimental results and concluding remarks giving future research directions are also included in this chapter.

2 Sensor Integration

2.1 Techniques and Approaches

Multi-sensor integration and fusion have been used in the last years in many areas. Typical applications are industrial tasks like assembly, multi-target tracking and aircraft navigation. The use of sensors with different characteristics is also common in mobile robotics, and it is well known that integration and fusion techniques allow to largely improve the performance of the overall system. The problem in this research area is to find strategies to integrate the information coming from different sensors in order to assist a system in the accomplishment of a task. This topic can be achieved either by using the information from one sensor to provide cues or guide the operations of the other sensors, or by fusing the information from multiple sensors.

Integration and fusion approaches have usually different aims. In particular, the sensor integration concept is mainly used to generically indicate the combined use of information from sensors of different types, no needing a common data representation for different sensors. Sensor fusion is carried out at a lower level, with the specific purpose of combining information from different types of sensors, usually with different data structure, into a single data representation, and with a degree of accuracy higher than single sources one. In particular, sensor fusion can be carried out both at the level of consecutive sets of data from the same sensor, and at the level of data from different sensors (see, among many others, [35], [1] and the references therein).

In other words the multi-sensor integration consists of the synergistic use of the information provided by multiple sensory devices in order to achieve a particular task; the multi-sensor fusion consists of one stage in the integration process where data from different sensors are combined into one representational format.

Clearly the overall architecture of the system strongly depends on the specific sensor and on the particular form of the information it provides; moreover it depends on the particular task for the entire system. Because of this variety of factors it is very hard to provide general methods able to enclose very different aspects of multi-sensor integration. Therefore, the results available in literature consist of a great amount of frameworks, paradigms sometimes experimentally tested. Many of these paradigms come from computer science, control theory, systems analysis and artificial intelligence. Among these

frameworks, the notions of *modularity*, *hierarchical structures* and *adaptability* have a particular importance.

The modularity of the integration process design reduces system complexity and increases its flexibility and possibility of being modified in relation to sensors kind and features. This is typical of structured programming.

Usually, the structure of the information processing system is intrinsically hierarchical. Data from the sensors are transformed into representation progressively more abstract, moving from the low level of the sensors up to the higher level, where information may become symbolic to represent complex concepts such as moving obstacles, target positions, landmarks and so on. This corresponds also to different control and planning levels in a navigation architecture. For example, the feedback motors control is usually based on current position measurements/estimates, while obstacle avoidance algorithms consider data at higher levels, and trajectory planning may take into account information at a further abstract level.

Finally integration process adaptability can be an efficient mean to overcome multi-sensor integration errors and uncertainties.

The basic advantages of sensor fusion and integration are: redundancy, complementarity, timeliness and cost. As a matter of fact, data from several sensors, whenever related to the same portion of the environment, are naturally redundant, and may allow uncertainty reduction. Similarly, if each type of sensor is specifically designed for a different feature, then the natural complementarity allows to reduce overall error. Finally, timeliness allows a faster data processing, while fusion and integration of low cost sensors may turn out in an overall sensing system with a good cost/performance ratio.

The role of multiple sensors in the operation of a particular system can then be defined as the degree to which each of these four aspects is present in the information provided by the sensors.

Data fusion has a great importance in mobile robotics. It is fundamental in order to integrate a localisation method that does not use beacons or markers, with a map-building algorithm that builds and constantly updates the world model (for more details see [2], [25], [17] and references therein).

It is important to consider the problems related to a given multisensor integration architecture. Because of the absence of a general methodology for multisensor integration and fusion (it in fact strongly depends both on the particular task to be achieved and on the sensors used), any integration architecture requires a careful planning phase. It is therefore necessary to analyse in detail the mobile robot (maximum speed, dimensions, vehicle model and so on) and the environment where it moves (indoor/outdoor characterisation), in order to find a good solution. Moreover, the architecture must manage data in a fast and efficient way because of the increased number of data, due to the increased number of sensors. Before sensor data can be fused, it is very important to make sure that these sensor data are consistent, in other words it is necessary to address the topic of sensor *closeness* [46]. Finally the

results obtained by many integration architectures are based on particular assumptions about how the sensors interact with the environment. In many contexts, as in outdoor environment, this interaction is hardly predictable.

A quite general model for the multi-sensor integration process is shown in Figure 2.1, where a general case with n sensors is considered [35]. In the following, S_i denotes the i^{th} sensor.

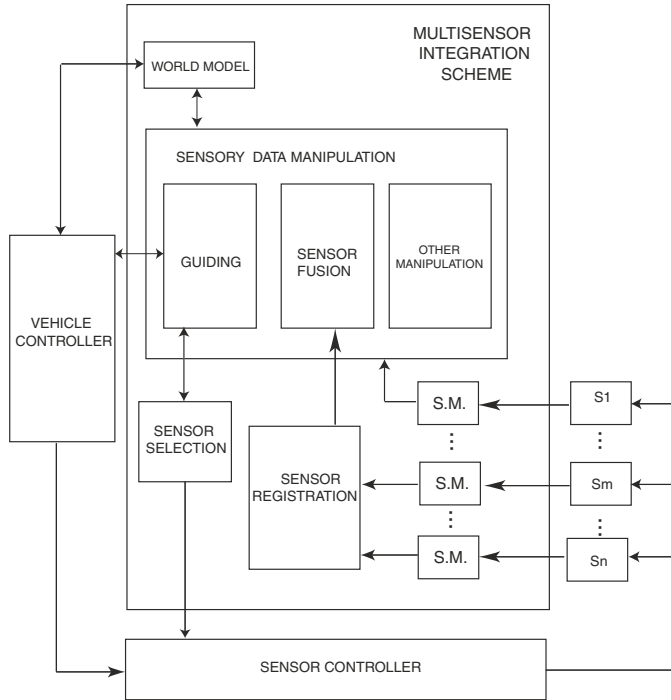


Figure 2.1. Multisensor integration

Clearly any integration architecture needs to know the information accuracy of each sensor in order to weigh all the data in a proper manner. In other words, a sensor model (indicated in the figure with S.M.) is required. After data have been modelled they can be integrated. It is possible to distinguish three different types of sensory processing [36]: fusion, guiding and other manipulation.

In the figure the data corresponding to sensors from S_m to S_n are fused. In order to do it, the data must be firstly made commensurable. Sensor registration consists of any data operation finalised to test data closeness, i.e. to understand if data refer to the same location in the environment over the same time period or not. Let X and Y be two sensory data vectors generated by two sensors and S_X and S_Y their covariance matrices. In order to test their

closeness a distance function must be introduced. Usually the Mahalanobis distance is used as matching criterion [53], [52], [46]. It is defined as follows:

$$d^2(X, Y) = (X - Y)^T (S_X + S_Y)^{-1} (X - Y). \quad (2.1)$$

In the case data provided by more than two sensors must be fused, it is necessary to have an algorithm able to determine which sensors refer to the same feature (reliable sensors) and which not (unreliable sensors). The Single Linkage Algorithm (SLA) is a useful way to achieve this result [46], by properly grouping sensors (*sensor clustering*).

A guiding type of sensory processing refers to the situation where the data from one sensor is used to guide the operation of other sensors. A typical example of this type of multi-sensor integration is found in many robotics applications where visual information is used to guide the operation of a tactile array mounted on the end of a manipulator.

2.2 Outdoor and Indoor Characterisation

Most of the methods for autonomous mobile robot navigation have as a preliminary step the recognition of landmarks in the environment, whose position is known. The robot must be able to recognise the landmarks using its sensors. A good landmark should be easily identified. In order to simplify the problem of landmark identification one supposes that the robot position and orientation are known with some accuracy. In this way the robot can reduce significantly the exploration area.

It is possible to distinguish two kinds of landmark: *natural landmark* and *artificial landmark*. The artificial landmarks are particular markers or objects introduced in the environment with the unique purpose of allowing autonomous navigation. On the contrary, the natural landmarks are objects or features already present in the environment, independently of the robot navigation. Clearly natural landmarks are more effective in an indoor environment, since it is easier to recognise particular geometrical features. This is achieved by different methods depending on the sensor used (for example Hough transform is very popular to detect lines from video images).

Concerning the navigation in an outdoor environment, the relocalisation method drastically changes. Most of the times means like the Global Positioning Systems (GPS) or compass are used but in some applications it is not possible (for example in space applications). In this case it becomes very important to improve the odometric localisation accuracy [9].

3 Sensor Fusion

In a mobile robotics context, sensor fusion consists in the direct integration of sensory data finalised to estimate robot parameters and/or state. The estimates obtained will be provided to motion planning and control algorithms.

In this section a classification of the different sensor fusion techniques together with some illustrative examples is reported.

The classification presented in this section follows the survey paper in [27]. It is based on the statistics knowledge about sensors and environment (known or unknown) and on fusion algorithm architecture (centralised or decentralised).

3.1 Sensor Fusion With Known Statistics in a Centralised Architecture

When the sensor system and the mobile robot environment statistics are known, the sensor fusion problem is well posed and, if a centralised architecture is used, the solution can be derived using well developed techniques based on the Kalman Filter (KF) or the Extended Kalman Filter (EKF). Game theory and Bayesian team theory are also used. Such techniques derive from the application of Maximum A Posteriori (MAP) and Maximum Likelihood (ML) estimation methods.

Assume the robot is provided with a set of sensors. They can be seen as members of a team whose purpose is to determine a parameter $x \in \mathbb{R}^n$. Sensors give noisy measures $y_i \in \mathbb{R}^p$, $i = 1, \dots, N$, related to the parameter x .

Consider the case x is a constant parameter: the case x is a time varying quantity (such as the case of state estimation) will be considered in the localisation section.

The recursive KF is first introduced following the approach in [4] in the case the i^{th} measurement y_i depends on the parameter x according to an equation of the following type:

$$y_i = C_i x + u_i, \quad (3.1)$$

where u_i is a noise component whose statistics satisfy:

$$E[u_i] = 0$$

$$E[u_i u_i^T] = U_i \geq 0$$

$$E[u_i u_j^T] = 0, \quad i \neq j.$$

These are reasonable assumptions, since it is always possible under proper transformations, to have a similar scenario [4]. The KF, if no Gaussian assumption is taken on the noise, provides the best (i.e. with minimum variance) linear unbiased estimate of x . This means that among all the estimators which seek \hat{x}_k as a linear combination of the measurements $\{y_i\}$, it minimises the expected error squared norm, i.e.

$$E[(\hat{x}_k - x)^T(\hat{x}_k - x)]$$

with $E[\hat{x}_k] = x$. Initially an estimate \hat{x}_0 of x is given together with its covariance matrix S_0 . The estimation process evolves according to the following set of equations:

$$\hat{x}_i = \hat{x}_{i-1} + K_i(y_i - C_i\hat{x}_{i-1}) \quad (3.2)$$

$$K_i = S_{i-1}C_i^T(U_i + C_iS_{i-1}C_i^T)^{-1} \quad (3.3)$$

$$S_i = (I - K_iC_i)S_{i-1}. \quad (3.4)$$

The term $(y_i - C_i\hat{x}_{i-1})$ is called the *innovation*: the new estimate \hat{x}_i is obtained from the previous estimate by adding an amount proportional to the innovation. The proportionality factor K_i is the *Kalman gain*. The final estimate \hat{x}_N will minimise:

$$Q = (x - \hat{x}_0)^T S_0^{-1} (x - \hat{x}_0) + \sum_{i=1}^N (y_i - C_i x)^T U_i^{-1} (y_i - C_i x), \quad (3.5)$$

and S_N represents its covariance matrix, i.e.

$$S_N = E[(\hat{x}_N - x)(\hat{x}_N - x)^T].$$

Under a Gaussian assumption on the noise, also the estimate \hat{x}_N will be Gaussian with $\hat{x}_N \sim N(x, S_N)$. In this case the KF provides the best unbiased estimate among all (i.e. even non linear) estimators.

Consider now the general case, where the dependence of the measurements on the parameter x to be estimated is not of the type (3.1). In this case it is possible to use the EKF. Let $\hat{z}_i \in \mathbb{R}^m$, $i = 1, \dots, N$ be the measurement of some unknown quantities z_i and assume that the relation between each one of these quantities and the parameter x is given by:

$$f_i(z_i, x) = 0, \quad i=1, \dots, N, \quad (3.6)$$

where f_i is a function from $\mathbb{R}^m \times \mathbb{R}^n$ into \mathbb{R}^p . Equation (3.1) is a particular case of this equation. The measurement \hat{z}_i is affected by some additive noise w_i :

$$\hat{z}_i = z_i + w_i, \quad (3.7)$$

where w_i is a random error with zero mean, known covariance and white noise:

$$E[w_i] = 0$$

$$E[w_i w_i^T] = W_i \geq 0$$

$$E[w_i w_j^T] = 0, \quad i \neq j.$$

It is possible to linearise the (3.6) if an estimate \hat{x}_{i-1} of parameter x based on the first $i-1$ measurements and an initial estimate \hat{x}_0 is already available. A first-order Taylor expansion of $f_i(z_i, x)$ performed around $(\hat{z}_i, \hat{x}_{i-1})$, which is a known point since \hat{z}_i is the i^{th} measurement, gives:

$$f_i(z_i, x) = 0 \cong f_i(\hat{z}_i, \hat{x}_{i-1}) + \frac{\partial f_i}{\partial z} \big|_{(\hat{z}_i, \hat{x}_{i-1})} (z_i - \hat{z}_i) + \frac{\partial f_i}{\partial x} \big|_{(\hat{z}_i, \hat{x}_{i-1})} (x - \hat{x}_{i-1}). \quad (3.8)$$

This equation can be written like equation (3.1) by setting:

$$\begin{aligned} y_i &= -f_i(\hat{z}_i, \hat{x}_{i-1}) + \frac{\partial f_i}{\partial x} \big|_{(\hat{z}_i, \hat{x}_{i-1})} \hat{x}_{i-1} \\ C_i &= \frac{\partial f_i}{\partial x} \big|_{(\hat{z}_i, \hat{x}_{i-1})} \\ u_i &= \frac{\partial f_i}{\partial z} \big|_{(\hat{z}_i, \hat{x}_{i-1})} (z_i - \hat{z}_i). \end{aligned}$$

Observe that y_i can be considered as a measurement, since it is known once \hat{z}_i is known. Also C_i is a known matrix. The quantity u_i is a noise component whose statistics depend on the statistics of the original noise w_i in the following way:

$$\begin{aligned} E[u_i] &= 0 \\ U_i &= E[u_i u_i^T] = \frac{\partial f_i}{\partial z} \big|_{(\hat{z}_i, \hat{x}_{i-1})} W_i \frac{\partial f_i}{\partial z}^T \big|_{(\hat{z}_i, \hat{x}_{i-1})}. \end{aligned}$$

If a Gaussian assumption is taken on the w_i then it also holds for the u_i . Once the approximation of equation (3.8) has been considered, the estimation procedure can be carried out through the KF procedure presented above.

The use of KF for robot navigation is considered in many papers like, for example, [3], [4], [15], [16], [22], [28], [33], [38], [29].

The KF for sensor fusion is a valid framework which can handle data coming from many and very different sensors, as long as assumptions for its applicability can be satisfied and sensors statistics are known and reliable. If sensor statistics are unknown, different techniques must be used, as will be explained in the relative section.

3.2 Sensor Fusion With Known Statistics in a Decentralised Architecture

In the previous section a centralised data fusion has been considered, i.e. the KF algorithm is executed by a single central processor, which processes the data coming from all the sensors and carries out all the computation to fuse them.

Now, the attention will be focused on decentralised architectures. Decentralisation is needed when the complexity of the mobile robot system requires computation and communication loads which can be carried out more efficiently with a parallel architecture. In addition, decentralisation and parallel computation improves the reliability of the overall system. It is assumed that the system comprises several sensors. In this case the fusion algorithm would require a lot of time to be executed on a single processor architecture.

To illustrate how a sensor fusion can be performed in a decentralised architecture, the implementation of the decentralised KF presented in [18] will be discussed. The architecture comprises several nodes, one for each sensor. Each sensor node is equipped with a processing unit and with communication facilities. It can perform autonomously its computation and communicate with any connected node exchanging computation results. The main features of this approach are:

- the estimates obtained by each node are exactly the same estimates which would have been obtained using a centralised architecture;
- the system is robust with respect to sensor failures;
- the amount of additional computation is small compared to the computation saved by the parallel architecture;
- the communication overhead is also low.

At the beginning of the algorithm, each node is given an estimate of the state of the system. The associated variance of this estimate is also available to the node. In general, data associated with the sensor of a given node have only a partial correspondence with the overall state of the system. In the main loop of the algorithm, each node takes an observation of the part of the state it can have access to. If the variance of these observations is known, the node can combine the previous estimate with the observation using the KF, to obtain a new partial estimate of the state. At this point, the nodes exchange their partial information. Each node, when data from other nodes are available, computes some assimilation equations to obtain a global state estimate. More details on this algorithm can be found in [18].

3.3 Sensor Fusion With Unknown Statistics

It is possible that the uncertainty models of sensor measurements are not well defined. This is especially true for sensors which are significantly different. However, it is often beneficial to use very different sensors for information complementarity purposes. In these cases, where uncertainty models are either unavailable or meaningless, the fusion is realised by some ad hoc techniques. In the following, some different approaches to this problem are illustrated.

A first possible approach is the so-called *rule-based sensor fusion*. It is based on the definition of a set of rules which allows to handle data coming from the sensors. The main disadvantage of this method is that such a set of rules is related to a specific environment and it cannot be easily used in other environments. In particular, a substantial change in the rules is required if changes occur in the environment or in the sensor system.

To explain this kind of approach, an example from [20] about fusion of sonar and infrared sensors will be described. Now, ultrasonic sensors give quite accurate distance measurements, even if they are characterised by wide

angles. On the other hand, infrared sensors have excellent angular resolution but provide poor depth measurements. Combining properly infrared and ultrasonic sensors could provide good measurements in both angular and depth position. The following rules given in [20] allow to handle data when conflicting information is obtained:

1. Do not consider the near-infrared sensor if the distance given by the sonar is greater than the maximum distance observable by the near-infrared sensor.
2. If the sonar gives the maximum value it can detect, then the real distance is greater.
3. If the infrared sensor goes from no-detection to detection, and the sonar sensor gives a distance in the range of the infrared sensor, a depth discontinuity has been found.

A typical application of rule-based sensor fusion is in map-assisted positioning. Other fusion approaches have been developed in the case measurements statistics are unknown. Some are an extension of the KF method [22]; others generalise the Bayesian approaches, in particular toward Shafer-Dempster reasoning [8], [21], [23]. Related to this problem are also [5], [34], [37], [26].

A combination of sensor fusion with known statistics and rule-based sensor fusion for navigation purposes is presented in [51].

4 Localisation

4.1 The Localisation Problem

One of the key issue in autonomous robot design is localisation. A very crude solution to such a *localisation problem* is dead-reckoning, that is, integration of the velocity information over time. The method suffers from severe stability limitations, e.g. due to integration error and wheel slippage, and therefore cannot be used alone for medium/long distance. The usual approach is to integrate this simple technique with other approaches, to achieve a better overall accuracy. The sensor fusion techniques described in a previous section are frequently used to this purpose.

To introduce a formal definition of the localisation problem, a unicycle-like reference robot is introduced, moving on a planar environment. It is possible to define the *robot configuration* (often referred to also as the *robot pose*) with respect to a global reference frame R by the vector $x_r = [x, y, \theta]^T$, containing its position and orientation. Without considering odometry errors the motion equations are given by the well known unicycle equations:

$$\dot{x} = \nu(t) \cos(\theta) \quad (4.1)$$

$$\dot{y} = \nu(t) \sin(\theta) \quad (4.2)$$

$$\dot{\theta} = \omega(t) \quad (4.3)$$

where $\nu(t)$ and $\omega(t)$ are the translational and rotational velocity of the robot. The above continuous time model can be approximated via a discrete time one, if the period T between two discrete time instants is small enough, and both $\nu(t)$ and $\omega(t)$ can be considered constant during such an interval, i.e. $\nu(t) = \nu(kT) = \nu(k)$, $t \in [kT, (k+1)T)$ and $\omega(t) = \omega(kT) = \omega(k)$, $t \in [kT, (k+1)T)$. Then, by integration of the previous unicycle equations it follows:

$$x(k+1) = x(k) + T\nu(k) \cos\left(\theta(k) + \frac{\omega(k)T}{2}\right) \quad (4.4)$$

$$y(k+1) = y(k) + T\nu(k) \sin\left(\theta(k) + \frac{\omega(k)T}{2}\right) \quad (4.5)$$

$$\theta(k+1) = \theta(k) + \omega(k)T. \quad (4.6)$$

The dead-reckoning approach is just the use of the previous equations, based on knowledge of translational and rotational velocities. Actually, the approach is affected by several types of error, such as integration errors, wheel slippage, bias and wheel irregularity, just to cite a few. This can be described by the following discrete time model:

$$x(k+1) = x(k) + T\nu(k) \cos\left(\theta(k) + \frac{\omega(k)T}{2}\right) + v_x(k+1) \quad (4.7)$$

$$y(k+1) = y(k) + T\nu(k) \sin\left(\theta(k) + \frac{\omega(k)T}{2}\right) + v_y(k+1) \quad (4.8)$$

$$\theta(k+1) = \theta(k) + \omega(k)T + v_\theta(k+1) \quad (4.9)$$

where the sequence $v(k) = [v_x(k) \ v_y(k) \ v_\theta(k)]^T$ is a random sequence (noise) modelling the errors. Vector $v(k)$, in the general case, has no special statistical properties. This leads easily to the conclusion that the above approach can be used only over a short distance path, since over long distance paths errors accumulate over time and quickly lead to divergence of the robot position estimation error.

Autonomous robots are usually equipped with on board sensing capabilities, allowing them to take heterogeneous measurements relating robot pose and the environment. Let $y(k)$ denote the generic vector of measurement taken at time k .

A common approach to the localisation problem is the use of a suitable estimation algorithm Λ , based on such a set of measurements $y(k)$:

$$\hat{x}_r(k+1) = \Lambda(\hat{x}_r(k), y(k)), \quad (4.10)$$

which provides the pose estimate minimising a suitable function of the estimation error $\tilde{x}_r(k)$, $\tilde{x}_r(k) := \hat{x}_r(k) - x_r(k)$.

A localisation algorithm can be seen as a procedure to align a *local map*, that is the environment as perceived by the mobile robot with its own sensors, with a *global map*, that is an independent, usually *a priori*, description of the

environment, available to the robot [45]. Then, several methods can be used to determine the optimal alignment between the two maps, hence the optimal estimation of the robot pose. As an example, in the case of localisation based on EKF-like sensor fusion, the measurement vector $y(k)$ is a description of the environment as perceived by the robot (i.e. the local map information), while the use of the (dynamic) model underlying the estimation phase corresponds to the use of the global map (or, an *a priori* knowledge of the environment) and a good alignment between the two maps is given by the EKF.

Of course, different approaches can be cast in such a framework. For example, in [45] a method is proposed, which makes use of the local and global map concepts. The maximum-likelihood similarity measure is used, in order to compare the two maps, and determine the optimal pose estimation. The approach used to identify the best position is a hierarchical divide-and-conquer algorithm, spanning the *pose space*. The approach turns out to be relevant in outdoor applications, and has been motivated by long-range rover projects for space missions.

To conclude the section, a quick mention of different variants of the localisation problem is given. The problem has been studied both for indoor and outdoor environments. At the same time, approaches have been proposed that make use of beacons or artificial landmarks, properly placed in the environment, and approaches only based on natural features. Interaction with the environment, i.e. sensing of the local map, can be carried out by means of sonars, laser finders, camera (monocular, stereo and omnidirectional), as well as inertial navigation systems, the GPS and similar equipment.

Here, a description of relevant schemes mostly for indoor environment, and based on natural features is given. Additional schemes can be found in [49], [45], [19], [42], [2] and the references therein. Other relevant schemes that will be briefly described in the following are based on the use of artificial landmarks and of nonlinear observers.

Finally, although attention here is mainly given to single robot localisation, the issue is under investigation also for the case of multi-robot systems, e.g. in case of collaborative localisation.

4.2 The Extended Kalman Filter

One of the most popular approaches to pose estimation, i.e. robot localisation, relays on the use of (Extended) Kalman filtering. The use of Kalman filtering for sensor fusion, henceforth for localisation, assumes the availability of a map of the environment. In the following such a map, which plays the role of the global map mentioned earlier, will be also referred to as the *localisation map* [39]. It is used during the localisation process to compute the predicted sensors values.

In the following it will be considered the case (widely treated in literature) of sonar sensors. Given a robot configuration x_r , the sonar expected value in x_r must be known. Consider an indoor bi-dimensional environment described

through straight segments representing the main walls. It is possible to use as prediction function for sonar S_i , modelled as a point $(s_{x,i}, s_{y,i})$, the minimum Euclidean distance between S_i and a point, $X = (x_i, y_i)$, belonging to a wall, so that the segment connecting S_i with X is contained in the i^{th} sonar beam-width. The predicted reading for the i^{th} sonar is then given by:

$$h_i(k, x_r) = \sqrt{(s_{x,i} - x_i)^2 + (s_{y,i} - y_i)^2}. \quad (4.11)$$

In the implementation of the EKF a threshold is often introduced: it depends on the distance covered by the robot during one cycle time and on the sonar noise; when the distance between a sonar data and the relative sonar prediction exceeds the threshold, the sonar data is considered incorrect and ignored in the localisation process. On the other hand all the sonar data can be used to update a map describing obstacles positions.

Denote by $y(k)$ the vector containing sonar data at time k . These data depend both on the robot configuration and on the environment, according to (4.11):

$$y(k) = h(k, x_r) + w(k) \quad (4.12)$$

where $w(k)$ (the sonar error) is assumed a zero-mean, white, Gaussian measurement noise with covariance matrix $R(k)$.

Let $\hat{x}_r(k|k) := [\hat{x}(k|k) \quad \hat{y}(k|k) \quad \hat{\theta}(k|k)]^T$ denote the conditioned state configuration of the robot, i.e. the estimate at time k based on the first k vectors of sonar data.

The solution of the localisation problem based on the EKF consists in the design of an algorithm Λ providing a sequence of configuration estimates $\hat{x}_r(k+1|k+1)$ of the robot,

$$\hat{x}_r(k+1|k+1) = \Lambda(\hat{x}_r(k|k), y(k+1)). \quad (4.13)$$

One cycle of the estimation algorithm will therefore consist in mapping the estimate $\hat{x}_r(k|k)$ and the associated state error covariance matrix $P(k|k) = E\{\hat{x}_r(k|k) \hat{x}_r(k|k)^T\}$ into the corresponding variables at the next stage $\hat{x}_r(k+1|k+1)$ and $P(k+1|k+1)$.

The EKF approach to this problem is briefly described in the following. Basically, it consists in the iteration of the two phases of pose prediction and pose estimation.

4.2.1 Prediction Phase. Starting from the configuration $\hat{x}_r(k|k)$ estimated at time k using sonar data, the configuration at the time $k+1$, i.e. $\hat{x}_r(k+1|k)$, is predicted.

$$\hat{x}(k+1|k) = \hat{x}(k|k) + T\nu(k) \cos\left(\hat{\theta}(k|k) + \frac{\omega(k)T}{2}\right) \quad (4.14)$$

$$\hat{y}(k+1|k+1) = \hat{y}(k|k) + T\nu(k) \sin\left(\hat{\theta}(k|k) + \frac{\omega(k)T}{2}\right) \quad (4.15)$$

$$\hat{\theta}(k+1|k+1) = \hat{\theta}(k|k) + \omega(k)T. \quad (4.16)$$

The associated state error covariance matrix is

$$P(k+1|k) = F(k)P(k|k)F(k)^T + Q(k, l) \quad (4.17)$$

where $F(k)$ is the Jacobian of the motion equations (4.4)-(4.6) (written in compact matrix form as $x_r(k+1) = f(k, x_r(k)) + \nu(k+1)$) with respect to the robot configuration, that is:

$$F(k) := [\nabla_X f^T(k, X)]_{X=\hat{x}_r(k|k)}. \quad (4.18)$$

From the *localisation map* and the previous predicted configuration $\hat{x}_r(k+1|k)$ of the robot, relation (4.11) allows to predict the sonar measurement at time $(k+1)$ according to:

$$\hat{y}(k+1) = h(k, \hat{x}_r(k+1|k)). \quad (4.19)$$

4.2.2 Estimation Phase. The estimation phase is based on the computation of the estimation error, defined as the difference between the estimated measurements and the actual sonar readings:

$$\tilde{y}(k+1) := y(k+1) - \hat{y}(k+1), \quad (4.20)$$

whose covariance is

$$S(k+1) = H(k+1)P(k+1|k)H(k+1)^T + R(k+1) \quad (4.21)$$

where $H(k)$ is the Jacobian of the function $h(k, X)$ with respect to the robot configuration, i.e.

$$H(k+1) = [\nabla_X h^T(k+1, X)]_{X=\hat{x}_r(k+1|k)}. \quad (4.22)$$

The final step consists in the computation of the Kalman gain $K(k+1)$ in order to compute $\hat{x}_r(k+1|k+1)$ and its covariance $P(k+1|k+1)$. In our notation:

$$K(k+1) = P(k+1|k)H(k+1)^T S^{-1}(k+1) \quad (4.23)$$

$$\hat{x}_r(k+1|k+1) = \hat{x}_r(k+1|k) + K(k+1)\tilde{y}(k+1) \quad (4.24)$$

$$P(k+1|k+1) = [I - K(k+1)H(k+1)^T]P(k+1|k). \quad (4.25)$$

4.3 Adaptive Extended Kalman Filter

One of the problems arising in EKF use for sensor fusion is the selection of proper values for the input and measurement noise covariance matrices. Often, the magnitude of these matrices has an impact on the performance of the overall localisation scheme. Such a problem is particularly relevant whenever the robot moves in complex or quickly changing environments. A possible approach to overcome this problem is the use of adaptation mechanisms to estimate the covariance matrices. The Adaptive EKF proposed in [25] is along this line. Its design takes into account filter divergence and complexity of the implementation.

A different approach to the same problem is proposed in [49]; it is based on joint Bayesian estimation and Kalman filtering.

4.4 Integrating Vision and Sonar

The solution to the localisation problem is often based on the use of vision systems too. In addition to the approach known as visual feedback [40], [54], and which is outside the scope of this text, vision data can be integrated with other sensors, along the lines briefly outlined in the first section of this chapter. In this case, the use of artificial landmarks is common.

Among many others, the approach presented in [44], [6] uses an integration scheme suitable for open space movements, comprising a vision system to correct the robot pose in the proximity of critical points, such as in the proximity of doors that have to be crossed. Those critical points are labeled by means of landmarks. The use of beacons or artificial landmarks is quite common in structured environment [32], [41], [24], such as offices, academic laboratories, and appear to be reasonable also in other scenarios, such as hospitals or museums. One of the main advantages of artificial landmarks is that they can be used to encode in a unique manner (e.g. by using bar code landmark) specific location within the map, so that upon detection of a single landmark, the robot can very precisely determine its position and orientation.

One of the main problems raised by vision systems is computational complexity.

4.5 Observability and Localisation

The interpretation of the localisation problem as an observability one is interesting and yields nice solutions. In [7], [14] the problem is formulated in terms of the observability of a dynamical system, comprising both the mobile robot and a set of landmarks, together with the nonlinear output equations relating robot configuration and landmark positions to the actual measurements. The overall system turns out to be nonlinear, and suitable tools for nonlinear system analysis are used to study observability, and to design nonlinear observers solving the localisation problem. One of the key problems that has been considered, and for which a possible solution is given, is that of singularity-avoidance.

4.6 Simultaneous Localisation and Map Construction

In a localisation problem, it is possible to efficiently exploit sensor readings only if an updated map of the environment is available. Some approaches have been proposed, to carry out simultaneous localisation and map construction.

In [2], the problem is formulated as a joint estimation of the robot pose and the environment parameters. The environment map is modelled by a collection of straight line segments, each segment being characterised by a set of four parameters. Hence, the solution is an algorithm providing sequences of estimates of pose and map data. Sensory data are given by odometry

and sonar readings. To cope with the complexity of the overall problem, a decomposition approach has been chosen, whereby a localisation phase and a map updating phase are executed iteratively. Experimental activities show that the approach has adequate performance.

The same problem has been studied also in [13], where a laser rangefinder and a CCD camera are used to drive an EKF updating estimate of a state vector comprising both robot pose and environment characterisation.

4.7 Multi-robot Localisation

The localisation problem described above, and the corresponding solutions, are well suited for the case of a single robot. A more complex localisation problem can be formulated if a team of mobile robots has to be considered. In this case, a simple approach would be that of running, on each single robot, one of the algorithms described earlier, depending on the available sensory equipment.

Nevertheless, some form of cooperation or collaboration among the robots may lead to a better overall accuracy. In the general case, it may occur that the robots have an heterogeneous sensory system, some robots being equipped with simple and low cost systems, and some other ones being fitted out with more powerful and accurate equipment. This multi-robot localisation is a recent issue. At the best of the author knowledge, initial contributions on the subject are [31], [30], based on the concept of “portable landmarks”. Other interesting contributions have been given in [48], [47], and [11], although the latter one is only partially centered on localisation. The approach in [48], called by the authors *collective localisation*, is worth mentioning here, since it is based on the use of a centralised scheme, implementing a single KF, estimating the pose of each robot in the team. The scheme fuses the information provided by the sensors (possible heterogeneous) on board of the individual robots, taking into account relationship among the collected data. Then, based on this initial idea, a distributed algorithm is proposed, so that eventually properly modified Kalman filters are implemented on each robot. Experimental results are presented, showing the benefit of the approach.

4.8 Some Experimental Results

The experiments described in the following [39], have been carried out on the Nomad 150 platform (able to move with a maximum translational velocity of about 0.3 ms^{-1}) available at the Robotics and Automation Laboratory, Department of Computer Science, Systems and Production, University of Rome Tor Vergata. Related results have been also presented in [43], [50].

4.8.1 Sonar Sensors. The sonar ranging system consists of 16 independent channels. This system can give range information from 0.4 m to 6.5 m with 1% accuracy over the entire range. It measures the time of flight ranging

sensor based upon the return time of an acoustic signal. Each sonar sensor has a beamwidth of approximately 22.5°.

4.8.2 Results. The experiments were carried out moving the Nomad150 in the hallways of the Department at the University of Rome Tor Vergata (see Figure 4.1). From this figure it is possible to see that the path from the initial to the final position is chosen in order to minimise the covered distance. However, it is not difficult to modify the proposed algorithm in order to minimise a different cost function.

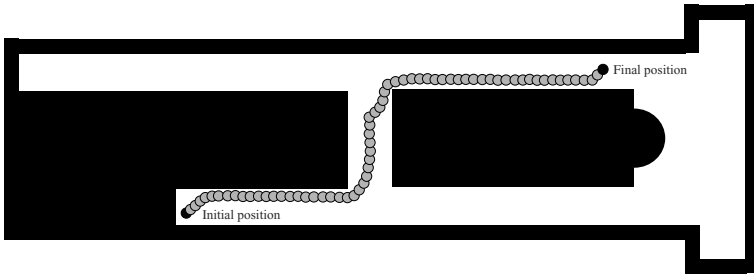


Figure 4.1. The map of the environment with the path of the robot. The path is chosen in this case in order to minimise the distance covered by the robot

The total cycle time for localisation (the most of that was due to radio-modem communication of sonars data) ranged from 0.7 s to 0.8 s.

In Figure 4.2 odometry and the estimated position against the actual position are compared. One may conclude that the proposed algorithm is able to correctly localise the robot, hence correcting the integration error of a pure dead-reckoning scheme.

The accuracy of the localisation process strongly depends on the translational velocity, since this process occurs while the robot is moving. The data shown in Figure 4.2 refer to the case where the localisation is performed when the robot is moving with the maximum translational velocity (0.3 ms^{-1}). It has been found that the error related to a motion with a speed of 0.1 ms^{-1} is about an half of the error showed in the figure.

5 Conclusion and Future Directions

In this chapter the use of a multi-sensor system for robot localisation has been considered. Robot localisation can be accomplished more efficiently using data coming from different sensors rather than data given by a unique sensor. The chapter focused on the techniques used to combine data from sensors with different characteristics.

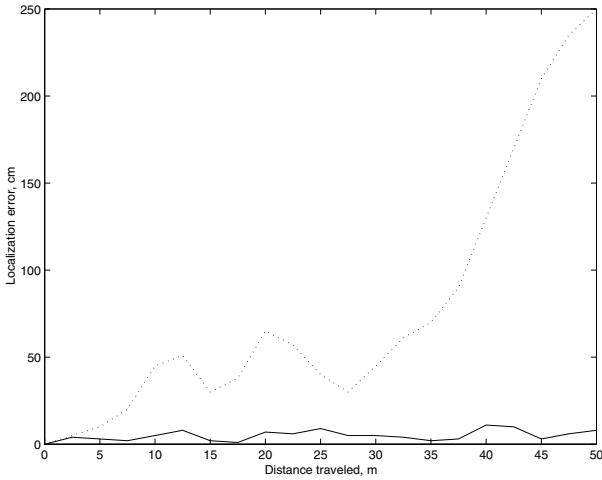


Figure 4.2. The error in the position of the robot during navigation estimated with odometry data alone (dotted line) and by the EKF localisation algorithm implemented (solid line). The translational velocity was 0.3 ms^{-1}

According to the particular environment considered (known or unknown), to the kind of sensors (with known or unknown statistics) and to the computational architecture (centralised or decentralised), different approaches can be used, which have been briefly considered in this chapter. A rich literature is related to this area of research. However, a vast array of open problems arises in this context. Some indications are provided in the following.

A standard approach in localisation problems is to compare data coming from sensors with their predicted values, determined using an environment map. However, it is not uncommon in real applications that the environment is not known or is not completely known. In this case a map must be constructed together with the localisation process. Some references on this subject are [2], [13].

A typical situation where a map of the environment is usually not available is in space exploration missions. This is an emergent area of research where new problems arise, also because some commonly used sensors like sonar can not be used. A related issue is the localisation of multi-robot teams. Different approaches can be considered in this case, as discussed in the localisation section.

References

1. Alag S, Goebel K, Agogino A 1995 A methodology for intelligent sensor validation and fusion used in tracking and avoidance of objects for automated

- vehicles. In: *Proceedings of 1995 American Control Conference* Seattle, WA, pp 3647–3653
2. Anousaki G C, Kyriakopoulos K J 1999 Simultaneous localization and map building for mobile robot navigation. *IEEE Transactions on Robotics and Automation* 15:42–53
3. Ayache N, Faugeras O D 1988 Building, registering and fusing noisy visual maps. *International Journal of Robotics Research* 7(6):45–65
4. Ayache N, Faugeras O D 1989 Maintaining representations of the environment of a mobile robot. *IEEE Transactions on Robotics and Automation* 5:804–819
5. Beom H R, Cho H S 1995 A sensor-based navigation for a mobile robot using fuzzy logic and reinforcement learning. *IEEE Transactions on Systems, Man and Cybernetics* 25:464–477
6. Bétourné A, Fefè A, Nicosia S, Valigi P 1997 A navigation system with self-localization capabilities. In: *Proceedings of 5th IEEE Mediterranean Conference on Control and Systems* Paphos, CY, pp 21–23
7. Bicchi A, Prattichizzo D, Marigo A, Balestrino A 1998 On the observability of mobile vehicles localization. In: *Proceedings of 6th IEEE Mediterranean Conference on Control and Systems* Alghero, I, pp 142–147
8. Bogler P L 1987 Shafer-Dempster reasoning with applications to multisensor target identification system. *IEEE Transactions on Systems, Man and Cybernetics* 17:968–977
9. Borenstein J 1998 Experimental results from internal odometry error correction with the OmniMate mobile robot. *IEEE Transactions on Robotics and Automation* 14:963–969
10. Brooks R 1991 Integrated systems based on behaviors. *SIGART Bulletin* 2:46–50
11. Burgard W, Moors M, Fox D, Simmons R, Thrun S 2000 Collaborative multi-robot exploration. In: *Proceedings of 2000 IEEE International Conference on Robotics and Automation* San Francisco, CA, pp 476–481
12. Butler G, Gantchev A, Grogono P 1999 Reusable strategies for software agents via the subsumption architecture. In: *Proceedings of 6th Asia Pacific Software Engineering Conference* Takamatsu, J, pp 326–333
13. Castellanos J A, Martínez J M, Neira J, Tardós J D 1998 Simultaneous map building and localization for mobile robots: A multisensor fusion approach. In: *Proceedings of 1998 IEEE International Conference on Robotics and Automation* Leuven, B, pp 1244–1249
14. Conticelli F, Bicchi A, Balestrino A 2000 Observability and nonlinear observers for mobile robot localization. In: *Preprints of 6th IFAC Symposium on Robot Control* Vienna, A, pp 115–120
15. Cox I J 1991 Blanche-An experiment in guidance and navigation of an autonomous robot vehicle. *IEEE Transactions on Robotics and Automation* 7:193–204
16. Crowley J 1989 World modeling and position estimation for a mobile robot using ultrasonic ranging. In: *Proceedings of 1989 IEEE International Conference on Robotics and Automation* Scottsdale, AZ, pp 674–680
17. Durrant-Whyte H, Pagac D, Nebot E M 1998 An evidential approach to map building for autonomous vehicles. *IEEE Transactions on Robotics and Automation* 14:623–629
18. Durrant-Whyte H F, Rao B Y S, Hu H 1990 Toward a fully decentralized architecture for multi-sensor data fusion. In: *Proceedings of 1990 IEEE International Conference on Robotics and Automation* Cincinnati, OH, pp 1331–1336

19. Fabrizio E, Oriolo G, Panzieri S, Ulivi G 2000 Mobile robot localization via fusion of ultrasonic and inertial sensor data. In: *Proceedings of International Symposium on Robotics with Application* Maui, HI, pp 121–127
20. Flynn A M 1988 Combining sonar and infrared sensors for mobile robot navigation. *International Journal of Robotics Research* 7(6):5–14
21. Garrey T, Lowrance J, Fischler M 1981 An inference technique for integrating knowledge for disparate sources. In: *Proceedings of 7th International Joint Conference on Artificial Intelligence* Vancouver, CAN, pp 319–325
22. Hong L, Wang G-J 1994 Integrating multisensor noisy and fuzzy data. In: *Proceedings of 1994 International Conference on Multisensor Fusion and Integration for Intelligent Systems* Las Vegas, NV, pp 199–206
23. Huntsberger T, Jayamamurthy S N 1987 A framework for multi-sensor fusion in the presence of uncertainty. In: *Proceedings of Workshop on Spatial Reasoning and Multi-sensor Fusion* St Charles, IL, pp 345–350
24. Janet A J, Luo R C, Kay M G 1997 Autonomous mobile robot global motion planning and geometric beacon collection using traversability vectors. *IEEE Transactions on Robotics and Automation* 13:132–140
25. Jetto L, Longhi S, Venturini G 1999 Development and experimental validation of an adaptive Extended Kalman Filter for the localization of mobile robots. *IEEE Transactions on Robotics and Automation* 15:219–229
26. Joshi R, Sanderson A C 1995 Multisensor fusion and unknown statistics. In: *Proceedings of 1995 IEEE International Conference on Robotics and Automation* Nagoya, J, pp 2670–2677
27. Kam M, Zhu X, Kalata P 1997 Sensor fusion for mobile robot navigation. In: *Proceedings of the IEEE* 85(1):108–119
28. Kleeman L, Kuc R 1995 Mobile robot sensor for target localization and classification. *International Journal of Robotics Research* 14:295–318
29. Kriegman D J, Triendl E, Binford T O 1989 Stereo vision and navigation in buildings for mobile robots. *IEEE Transactions on Robotics and Automation* 5:792–803
30. Kurazume R, Hirose S 1998 Study on cooperative positioning systems: Optimum moving strategies for cps-iii. In: *Proceedings of 1998 IEEE International Conference on Robotics and Automation* Leuven, B, pp 2896–2903
31. Kurazume R, Nagata S, Hirose S 1994 Cooperative positioning with multiple robots. In: *Proceedings of 1994 IEEE International Conference on Robotics and Automation* San Diego, CA, pp 1250–1257
32. Leonard J J, Durrant-Whyte H F 1991 Mobile robot localization by tracking geometric beacons. *IEEE Transactions on Robotics and Automation* 7:376–382
33. Leonard J J, Durrant-Whyte H F, Cox I J 1992 Dynamic map building for an autonomous mobile robot. *International Journal of Robotics Research* 11:286–298
34. Li W 1994 Fuzzy logic based robot navigation in uncertain environments by multisensor integration. In: *Proceedings of 1994 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems* Las Vegas, NV, pp 259–265
35. Luo R, Kay G 1989 Multisensor integration and fusion in intelligent systems. *IEEE Transactions on Systems, Man and Cybernetics* 19:901–931
36. Luo R C, Kay M G 1990 A tutorial on multi-sensor integration and fusion. In: *Proceedings of 16th IEEE Industrial Electronics Conference* Pacific Grove, CA, pp 707–722
37. Madsen D, Gunderson R 1994 An autonomous rover navigation scheme using fuzzy logic. In: *Proceedings of Joint Conference on Information Science* Pinehurst, NC, pp 204–207

38. Maeyama S, Ohya A, Yuta S 1995 Non-stop outdoor navigation of a mobile robot. In: *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems* Pittsburgh, PA, pp 130–135
39. Martinelli A, Martinelli F, Nicosia S, Valigi P 2000 Multisensor fusion for mobile robot positioning and navigation. In: *Preprints of 6th IFAC Symposium on Robot Control* Vienna, A, pp 409–414
40. Murphy R R 1998 Sensor and information fusion improved vision-based vehicle guidance. *IEEE Intelligent Systems* 13(6):49–56
41. Murhpy R R, Hersherberger D, Blauvelt G R 1997 Learning landmark triples by experimentation. *Robotics and Autonomous Systems* 22:377–392
42. Neira J, Tardós J D, Horn J, Schmidt G 1999 Fusing range and intensity images for mobile robot localization. *IEEE Transactions on Robotics and Automation* 15:76–84
43. Nicosia S, Capobianco A 1998 Controllo di robot mobili mediante il metodo del potenziale. *Automazione e Strumentazione* 46(9):117–124
44. Nicosia S, Valigi P 1998 A multi-sensor navigation system for mobile robots. In: *Proceedings of 6th IEEE Mediterranean Conference on Control and Systems* Alghero, I, pp 148–153
45. Olson C F 2000 Probabilistic self-localization for mobile robots. *IEEE Transactions on Robotics and Automation* 16:55–66
46. Raju G, Wang H 1994 Sensor data fusion using Mahalanobis distance and single linkage algorithm. In: *Proceedings of 1994 IEEE International Conference on Systems, Man and Cybernetics* San Antonio, TX, pp 2605–2610
47. Rekleitis I M, Dudek G, Milios E 2000 Multi-robot collaboration for robust exploration. In: *Proceedings of 2000 IEEE International Conference on Robotics and Automation* San Francisco, CA, pp 3164–3169
48. Roumeliotis S I, Bekey G A 2000 Collective localization: A distributed Kalman Filter approach to localization of groups of mobile robots. In: *Proceedings of 2000 IEEE International Conference on Robotics and Automation* San Francisco, CA, pp 2958–2965
49. Roumeliotis S I, Bekey G A 2000 Bayesian estimation and Kalman filtering: A unified framework for mobile robot localization. In: *Proceedings of 2000 IEEE International Conference on Robotics and Automation* San Francisco, CA, pp 2985–2992
50. Santini A, Nicosia S, Nanni V 1997 Trajectory estimation and correction for a wheeled mobile robot using heterogeneous sensors and Kalman Filter. In: *Preprints of 5th IFAC Symposium on Robot Control* Nantes, F, pp 11–16
51. Tarin C, Brugger H, Moscardo R, Tibken B, Hofer E P 1999 Low level sensor fusion for autonomous mobile robot navigation. In: *Proceedings of 16th IEEE Instrumentation and Measurement Technology Conference* Venice, I, pp 1377–1382
52. Weckesser P, Dillmann R 1997 Navigating a mobile service-robot in a natural environment using sensor-fusion techniques. In: *Proceedings of 1997 IEEE/RSJ International Conference on Intelligent Robots and Systems* Grenoble, F, pp 1423–1428, 1997
53. Weckesser P, Dillmann R, Rembold U 1996 Sensor fusion of intensity and laser-range-images, In: *Proceedings of International Conference on Multisensor Fusion and Integration for Intelligent Systems* Washington, DC, pp 501–508
54. Wei G-Q, Hirzinger G 1999 Multisensory visual servoing by a neural network. *IEEE Transactions on Systems, Man and Cybernetics — Part B* 29:276–280